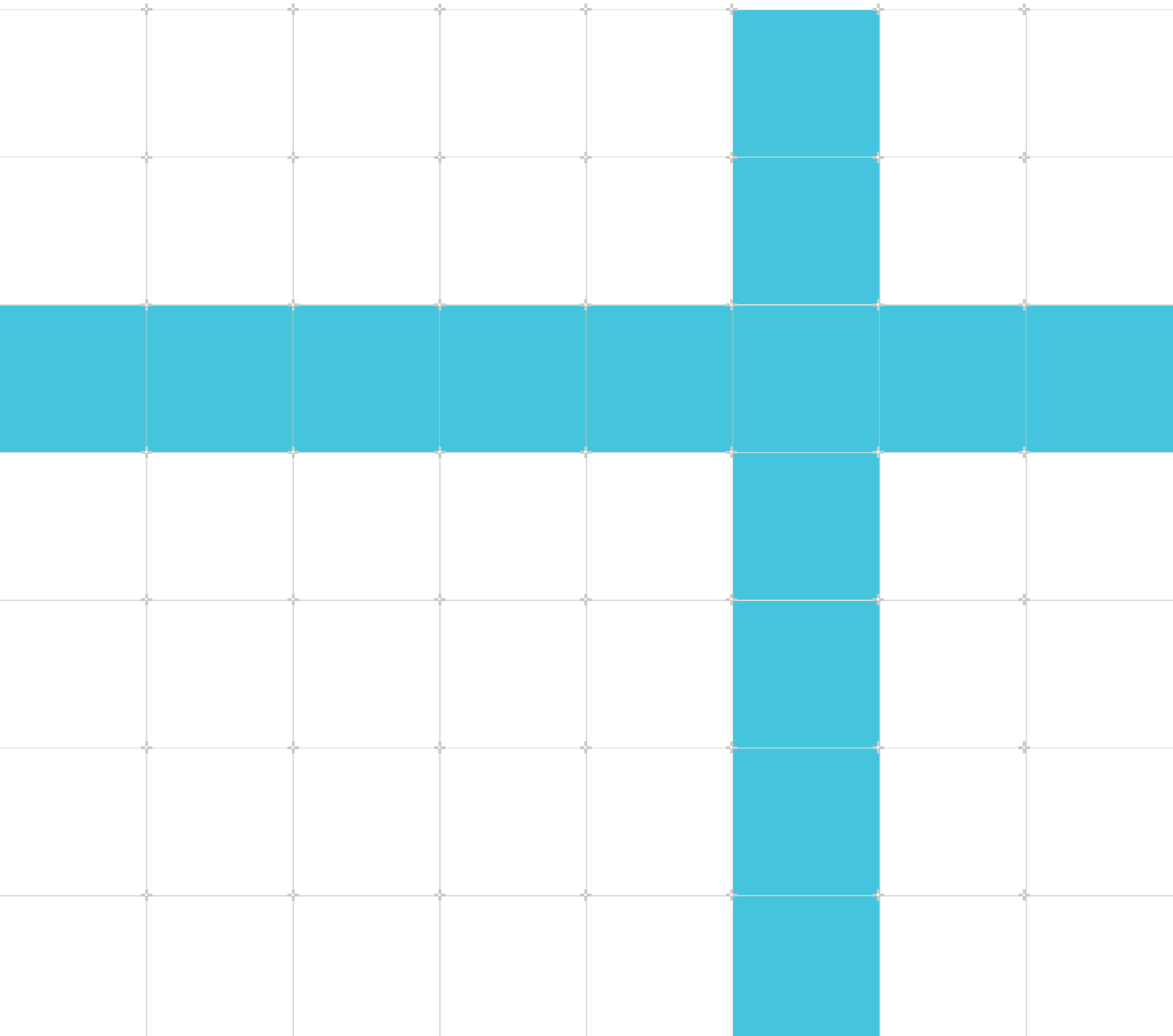
Arm® Neoverse™ CMN-700 Coherent Mesh Network

Revision: r3p2

Technical Reference Manual

Non-Confidential

Issue 07
102308_0302_07_en



Arm® Neoverse™ CMN-700 Coherent Mesh Network

Technical Reference Manual

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-01	22 December 2020	Confidential	First release for r0p0 LAC
0000-02	2 March 2021	Confidential	Second release for r0p0 LAC
0100-03	3 August 2021	Confidential	First release for r1p0 LAC
0200-04	17 November 2021	Confidential	First release for r2p0 LAC
0300-05	7 September 2022	Non-Confidential	First release for r3p0 EAC
0301-06	20 February 2023	Non-Confidential	First release for r3p1 EAC
0302-07	19 July 2023	Non-Confidential	First release for r3p2 EAC

Proprietary Notice

This document may include technical inaccuracies or typographical errors.

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

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

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

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.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com.

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

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

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Introduction

1.1 Product revision status

The r*x*p*y* identifier indicates the revision status of the product described in this manual, for example, r1p2, where:

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

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

1.2 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 Arm® Neoverse™ CMN-700 Coherent Mesh Network.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

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: developer.arm.com/glossary.

Convention	Use
italic	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

Convention	Use
<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	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

Recommendations. Not following these recommendations might lead to system failure or damage.

Requirements for the system. Not following these requirements might result in system failure or damage.

Requirements for the system. Not following these requirements will result in system failure or damage.

An important piece of information that needs your attention.

A useful tip that might make it easier, better or faster to perform a task.

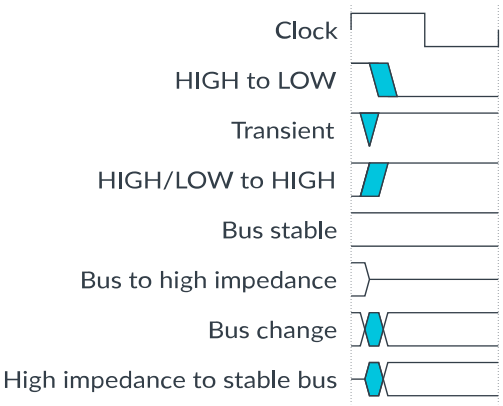
A reminder of something important that relates to the information you are reading.

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 1-1: 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, n denotes an active-LOW signal.

1.4 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual	102309	Confidential
Arm® Socrates™ User Guide	101399	Non-Confidential

Arm product resources	Document ID	Confidentiality
Arm® Socrates™ Installation Guide	101400	Non-Confidential
Arm® Neoverse™ N1 hyperscale reference design GIC-600 Integration using CMN-600 AXI4-Stream Interfaces White Paper [Only available on request]	PJDOC-1779577084-5931	Confidential
Arm® Neoverse™ CMN-700 Coherent Mesh Network Release Note	108084	Confidential

Arm architecture and specifications	Document ID	Confidentiality
AMBA® AXI and ACE Protocol Specification	IHI 0022H.c	Non-Confidential
AMBA® APB Protocol Specification	IHI 0024E	Non-Confidential
AMBA® APB Protocol Specification	IHI 0024D	Non-Confidential
AMBA® CXS Protocol Specification	IHI 0079A	Non-Confidential
AMBA® Low Power Interface Specification Arm® Q-Channel and P-Channel Interfaces	IHI 0068D	Non-Confidential
AMBA® 4 AXI4-Stream Protocol Specification	IHI 0051B	Non-Confidential
AMBA® 5 CHI Architecture Specification	IHI 0050E.c	Non-Confidential
Arm® CoreSight™ Architecture Specification	IHI 0029E	Non-Confidential
Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile	DDI 0487	Non-Confidential
Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A	DDI 0598	Non-Confidential
Principles of Arm® Memory Maps White Paper	DEN 0001	Non-Confidential

Non-Arm resources	Document ID	Organization
Standard Manufacturers Identification Code	*JEP106*	https://www.jedec.org
Compute Express Link (CXL) Specification Revision 3.0 - version 1.0 Release Candidate	-	https://www.computeexpresslink.org/

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

Adobe PDF reader products can be downloaded at http://www.adobe.com.

2. What is CMN-700?

This chapter introduces CMN-700 which is an AMBA® 5 CHI interconnect with a customizable mesh topology.

2.1 About CMN-700

The CMN-700 product is a scalable configurable coherent interconnect that is designed to meet the *Power, Performance, and Area* (PPA) requirements for Coherent Mesh Network systems that are used in high-end networking and enterprise compute applications.

CMN-700 is a scalable mesh interconnect with 1-256 processor compute clusters.

CMN-700 is configurable with the Arm® Socrates™ IP Tooling platform. Socrates™ is an environment for the configuration of Arm IP. Using Socrates™, you can configure the following CMN-700 characteristics:

- Custom interconnect size and device placement.
- Optional *System Level Cache* (SLC). For more information about the features of the SLC memory system, see 5.1 About the SLC memory system on page 1157.

CMN-700 supports Arm® AMBA® 5 CHI Issue E, including the following features:

- MakeReadUnique, writes with optional data, and write zero with no data transactions
- Enhanced Exclusive transactions
- Various transaction optimizations and enhancements
- Connection of devices with multiple interfaces
- Connection of devices with replicated channels
- Extended TxnID and GroupID
- *Distributed Virtual Memory* (DVM) updates
- Memory tagging

CMN-700 provides system-level alignment by providing the following system functionality:

- *Quality of Service* (QoS)
- *Reliability, Availability, and Serviceability* (RAS)
- *Debug and Trace* (DT)

CMN-700 is compatible with the following types of IP:

- *Dynamic Memory Controller* (DMC)
- *Generic Interrupt Controller* (GIC)
- *Memory Management Unit* (MMU)

- Interconnects such as the Arm® CoreLink™ NIC-450 Network Interconnect
- Armv8.2, Armv8.4, and Armv9.0 processors

CMN-700 provides an optional *Coherent Multichip Link* (CML) feature. CML is compliant with:

- CXL2.0 standard for memory expansion
- SMP connection

The following table shows the protocol nodes and devices that a system that is built using CMN-700 can contain:

Table 2-1: Supported protocol nodes and devices

Protocol node or device	Description
Fully coherent Requesting Node (RN-F)	A fully coherent manager device that supports: • CHI Issue B • CHI Issue C • CHI Issue D • CHI Issue E
Reliability, Availability, andI/O coherent Requesting Node (RN-I)	An I/O-coherent manager device. This CHI bridge device acts as an RN-I proxy for one or more AXI or ACE-Lite manager devices that connect to it.
I/O coherent Requesting Node with DVM support (RN-D)	An I/O coherent manager device that supports acceptance of *Distributed Virtual Memory* (DVM) messages on the Snoop channel
Fully coherent Home Node (HN-F)	A device that acts as a Home Node for a coherent region of memory. HN-Fs accept coherent requests from RN-Fs and RN-Is, and generates snoops to all applicable RN-Fs in the system as required to support the coherency protocol.
I/O coherent Home Node (HN-I)	A device that acts as a Home Node for the subordinate I/O subsystem, responsible for ensuring proper ordering of requests targeting the subordinate I/O subsystem. HN-I supports AMBA® AXI and ACE-Lite protocols.
I/O coherent Home Node + DVM Node (HN-D)	A device that includes an HN-I, a *Debug Trace Controller* (DTC), *DVM Node* (DN), *configuration node* (CFG), Global Configuration Subordinate, and the *Power/Clock Control Block* (PCCB). **Note:** Only one HN-D is allowed per CMN-700 instance. HN-D supports the AMBA® AXI, ACE-Lite, ATB, and APB protocols.
I/O coherent Home Node + DTC (HN-T)	An HN-I with a built-in DTC, *DVM Node* (DN), and ATB. HN-T supports the AMBA® AXI and ACE-Lite protocols.
I/O coherent Home Node + Distributed DVM Node (HN-V)	A device that includes an HN-I and *DVM Node* (DN). HN-V supports the AMBA® AXI and ACE-Lite protocols.
I/O coherent Home Node + PCIe optimization (HN-P)	A device that includes HN-I and dedicated trackers for PCIe peer-to-peer traffic. This device can only be used to connect to PCIe subordinates. HN-P supports the AMBA® AXI and ACE-Lite protocols.
CHI Subordinate Node (SN-F)	A device which solely receives CHI commands, limited to fulfilling simple read, write, and CMO requests targeting normal memory
AMBA 5 CHI to ACE5-Lite bridge SBSX	A CHI bridge device that converts and forwards simple CHI read, write, and CMO commands to an AXI or ACE-Lite subordinate memory device. Device ordering is not maintained in SBSX.

Protocol node or device	Description
CXL Gateway (CCG)	A CCG device bridges between CHI and CXS Issue B (SMP or CXL port)

2.2 Compliance

The CMN-700 product is based on Issue E of the *AMBA® 5 CHI Architecture Specification*.

This *Technical Reference Manual* (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

CMN-700 supports the *AMBA® 5 CHI Architecture Specification* Issue E, and is also backwards compatible with Issue D, Issue C, and Issue B. For more information about compatibility, see 3.1.1.18 Backward compatible RN-F support on page 55.

The CMN-700 product implements the following architecture capabilities:

- Fully compliant with CHI interconnect architecture
- Non-blocking coherence protocol
- Packet-based communication
- The following four types of channels:
  - *Request* (REQ)
  - *Response* (RSP)
  - *Snoop* (SNP)
  - *Data* (DAT)
- Credited end-to-end protocol-layer flow-control with a retry once mechanism for flexible bandwidth and resource allocation
- Integrated end-to-end *Quality-of-Service* (QoS) capabilities

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

CXL architecture

CMN-700 supports CXL.mem (Type3) functionality for host implementation and is compliant with *Compute Express Link (CXL) specification revision 2.0* dated October 2020

2.3 Features

CMN-700 provides the following key features:

- Highly scalable mesh network topology configurable up to a 12 × 12 mesh

- Custom mesh size and device placement
- A programmable *System Address Map* (SAM)
- Up to 256 RN-F interfaces for CHI-based compute clusters, accelerators, graphic processing units, or other cache coherent managers
- *Component Aggregation Layer* (CAL) for device interface port expansion
- Up to 64 SN interfaces, 128 with CAL
- Up to 90 RN-Is with up to three ACE5-Lite ports each (270 total)

More devices are supported by adding more levels of interconnect hierarchy to the system. For example, you can use the Arm® CoreLink™ NIC-450 Network Interconnect to add more levels of interconnect hierarchy.

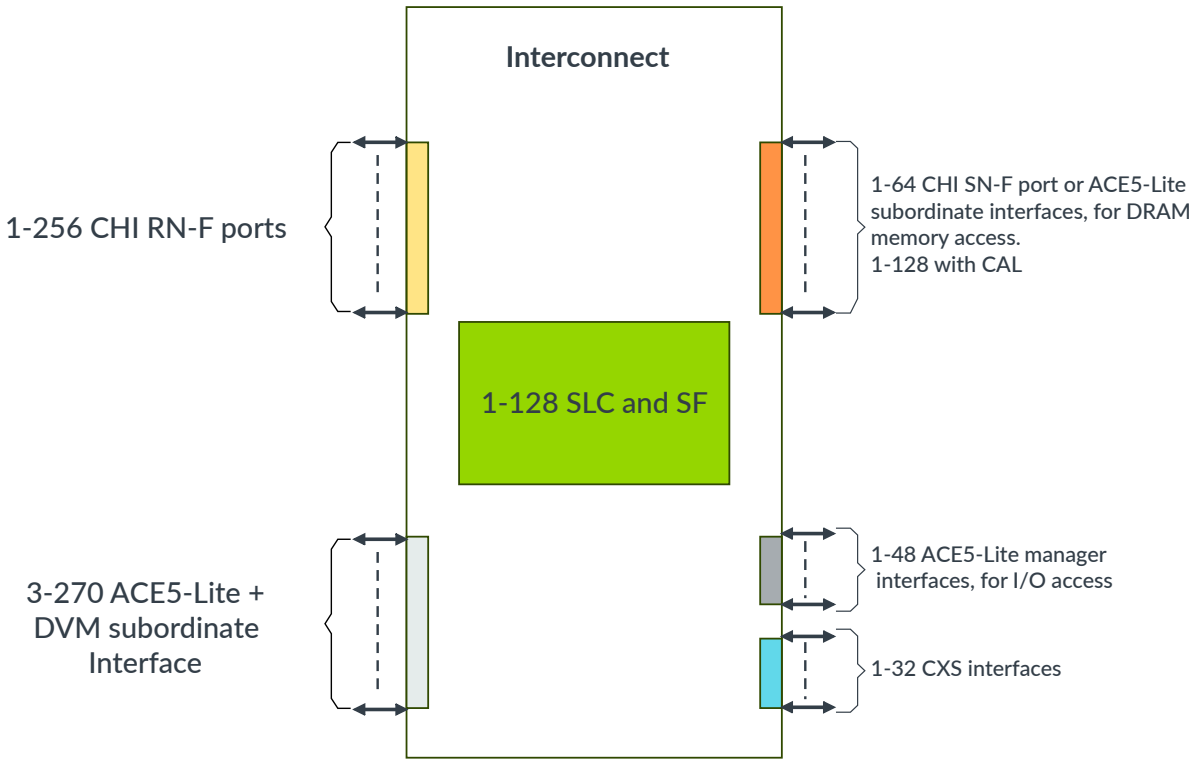
- Option for a second pair of RSP, REQ, SNP and 256-bit DAT channels, one for each direction. CMN-700 supports either single or dual RSP, REQ, SNP, and 256-bit DAT channel configurations for each direction.
- Optional support for non-XY routing algorithm between specified source-target pairs
- Maximum *Physical Address* (PA) width of 52 bits
- DVM message transport between managers
- QoS regulation for shaping traffic profiles
- Configurable QoS override to transactions targeting specific memory regions
- A *Performance Monitoring Unit* (PMU) to count performance-related events
- High-performance distributed SLC and *Snoop Filter* (SF) up to 128 HN-Fs and cache sizes of 0-512MB total:
  - The HN-F includes an integrated *Point-of-Serialization* (PoS) and *Point-of-Coherency* (PoC). The HN-F SLC (also referred to as Agile System Cache) can be used both for compute and I/O caching.
  - SF up to 1024MB of tag RAM for increased coherency scalability consisting of up to 128 partitions (one per HN-F).
- Up to 48, combinations of HN-Is (HN-T, HN-V, HN-P, HN-D), each with an ACE-Lite manager port
- An HN-I that is known as HN-P, which includes PCIe optimizations
- CHI *Memory Tagging Enhancements*
- Support for up to
  - 4 devices on corner *Mesh Crosspoint (MXPs)*
  - 3 devices on edge *MXPs*
- *AXI Utility Bus* (AXU) configuration interface for DSU and CHI SN-F interfaces
- *AXI4-Stream* (A4S) support, for GIC traffic only

- Option for a single synchronous clock domain or four rectangular asynchronous clock with the same clock frequency
- *Device Credited Slices* (DCSs) used for register slices at device interfaces, allowing flexibility in device placement.
- *Mesh Credited Slices* (MCSs) used for X-Y register slices, allowing flexibility in mesh floorplanning
- *Asynchronous Mesh Credited Slices* (AMCSs) used for asynchronous clock domain crossing
- *CAL Credited Slices* (CCSs), allowing flexibility in mesh floorplanning in configurations that use the CAL
- *On-Chip Memory* (OCM) allowing the creation of CMN-700 systems without physical DDR memory
- RAS features including transport parity, optional data path parity, *Single-Error Correction* (SEC) and *Double-Error Detection* (DED) ECC, and data poisoning signaling
- Up to 32 CCG devices
    - CCG devices support *Coherent Mesh Link SMP* (CML_SMP) or CXL2.0, and have 512-bit CXS Issue B interfaces
- *Address Based Flush* (ABF)
- Way-based SLC partitioning
- Source-based way locking
- CML Subordinate Agent support for CXL memory expansion
- Support for AXI read/write burst preservation on PCIe peer-to-peer traffic passing through interconnect targeting local or remote chip

2.4 Interfaces

The following figure shows the interfaces of the CMN-700 product.

Figure 2-1: CMN-700 interfaces



2.5 Configurable options

The basic structure of CMN-700 is a configurable rectangular grid that is composed of network routers that are known as *Crosspoints* (XPs) and CHI-compliant devices.

Each XP connects horizontally and vertically to other XPs, creating a two-dimensional mesh structure. In a mesh configuration, each XP can have up to four device ports for connecting CHI-compliant devices. See 3.1.1.1 Crosspoint on page 40.

CMN-700 provides several configurable parameters that can be configured to meet system requirements. You can use Socrates™ to refine the mesh design and device placement using the following guidelines.

CMN-700 is configured using the following steps:

1. System component selection. In this step, system components are determined, including:
   - Number and type of processors

- I/O interfaces
- Number of HN-Fs
- Amount of SLC
- Memory interfaces

See 2.5.1 System component selection on page 23.

2. Mesh sizing and top-level configuration. This step includes specifying the following:
   - Number of rows and columns
   - Global configuration parameters

   See 2.5.2 Mesh sizing and top-level configuration on page 25.
3. Device placement and configuration. This step involves:
   - Placement of devices and credited repeater slices between XPs based on floorplan requires
   - Configuration of devices

   See 2.5.3 Device placement and configuration on page 30.

2.5.1 System component selection

The following describes CMN-700 system component selection.

Request Nodes

Request Nodes (RNs) reside outside of the mesh and connect to CMN-700 ports.

Requesting managers with coherent caches (processors, GPUs, or processing elements with internal coherent caches) are referred to as RN-F devices. They connect directly to the CMN-700 interconnect mesh using a CHI RN-F port.

I/O-requesting managers without coherent caches connect to CMN-700 RN-I bridge devices using ACE-Lite ports. Examples of I/O-requesting managers include I/O managers, processing elements without internal caches, or processing elements with internal caches that are not hardware coherent. The RN-I bridge device is located between the ACE-Lite interface and the internal CHI interface. Each RN-I bridge device has three ACE-Lite interfaces.

A single I/O-requesting manager can be connected directly to a CMN-700 ACE-Lite port. Alternatively, multiple managers can share a single ACE-Lite port by connecting through external AMBA interconnect components. To determine if I/O managers share 1-3 ACE-Lite ports or an RN-I, designers must consider traffic bandwidth requirements and physical floorplan trade-offs.

Home Nodes

In CHI, each byte of address space is assigned to a single *Home Node* (HN). That HN is responsible for handling all memory transactions that are associated with that address.

There are two types of HN devices within the CMN-700 system:

HN-F

HN-F device instances are the HNs for all coherent memory. HN-Fs also support non-coherent memory accesses. Memory that is mapped to an HN-F targets DRAM. Each HN-F can contain an SF and an SLC slice. The amount of SLC required determines the number of HN-Fs.
The total SLC size required divided by the number of HN-F instances determines the recommended SLC size for each HN-F instance. Generally, each HN-F partition has the same SLC size. The amount of SLC and number of HN-Fs are configured separately.

The optimal total SF size is twice the total exclusive cache size for all RN-Fs. For example, for a 32MB RN-F total cache size, the recommended SF size is 64MB.

HN-I

HN-I device instances are the HNs for all memory that targets an ACE-Lite subordinate device or subsystem. HN-I does not support coherent memory. However, Cacheable transactions can be sent to HN-I. Each HN-I instance contains a single ACE-Lite manager port to send bus transactions to one or more subordinates through an AMBA interconnect. The total ACE-Lite manager bandwidth requirement, and physical placement of subordinate peripherals, determines the number of HN-I instances that are required.
There are HN-I types with extra functionality. These types include:

HN-T

HN-I that has a debug trace controller and *DVM Node*.
CMN-700 can have zero or more HN-I and HN-T instances.

HN-D

HN-I that has a debug trace controller, *DVM Node*, and configuration subordinate.
CMN-700 must have exactly one HN-D instance.

HN-P

HN-I that is optimized for peer-to-peer PCIe traffic.

HN-V

A device that includes an HN-I and *DVM Node* (DN).

CML interfaces

The CMN-700 interconnect supports up to 32 CXS (CXL/CML_SMP) interfaces. Each CCG device has a CXS interface, which is compliant with CXS Issue B specification.

A CCG device bridges between CHI and CXS, and contains *Request Agent* (RA) proxy and *Home Agent* (HA) proxy functionality. The CCG device also contains CXS *Link Agent* (LA) functionality, where:

- CCG LA implements CML_SMP/CXL flits and is internal to the CMN-700 hierarchy

Memory interfaces

The CMN-700 interconnect supports two types of memory interface ports:

CHI SN-F port

Connects a native CHI memory controller that complies with:

- CHI Issue C
- CHI Issue D
- CHI Issue E

ACE-Lite port

Connects an AXI or ACE-Lite memory controller using an SBSX. For more information, see 2.5.2 Mesh sizing and top-level configuration on page 25.

2.5.2 Mesh sizing and top-level configuration

The size of the CMN-700 mesh primarily depends on the number of connected devices.

The minimum number of XPs is half of the number of devices, rounded up. Also, the product of the X and Y mesh dimensions must be greater than or equal to the required number of XPs. For example, if seven XPs are needed, a 2×4 or 4×2 mesh would be acceptable.

The following table lists the device types that CMN-700 supports.

Table 2-2: Device types

Device	Name	Description
RN-I	Request Node I/O	A non-caching Request Node that bridges I/O manager requests from 1-3 AXI or ACE-Lite interfaces.
RN-D	DVM Request Node	An RN-I node that can accept DVM messages on the snoop channel.
RNF_CHIB_ ESAM	Request Node Full without a built-in SAM. CHI Issue B compliant.	CHI Issue B compliant processor, cluster, GPU, or other Request Node with a coherent cache but without a built-in SAM
RNF_CHIC_ ESAM	Request Node Full without a built-in SAM. CHI Issue C compliant.	CHI Issue C compliant processor, cluster, GPU, or other Request Node with a coherent cache but without a built-in SAM
RNF_CHID_ESAM	Request Node Full without a built-in SAM. CHI Issue D compliant.	CHI Issue D compliant processor, cluster, GPU, or other Request Node with a coherent cache but without a built-in SAM.
RNF_CHIE_ESAM	Request Node Full without a built-in SAM. CHI Issue E compliant.	CHI Issue E compliant processor, cluster, GPU, or other Request Node with a coherent cache but without a built-in SAM.
HN-F	Fully Coherent Home Node	A fully coherent Home Node, typically configured with one or both of SLC and SF.
HN-I	Home Node I/O	A device that acts as a Home Node for the subordinate I/O subsystem, responsible for ensuring proper ordering of requests targeting the subordinate I/O subsystem. HN-I supports AMBA AXI or ACE-Lite.

Device	Name	Description
HN-T	Home Node I/O with debug trace control	An HN-I with a built-in debug trace controller
HN-D	DVM Home Node	An HN-I with a built-in debug trace controller, *DVM Node* (DN), *Configuration Node* (CFG), Global Configuration Subordinate, and the *Power/Clock Control Block* (PCCB)
HN-P	Home Node I/O + PCIe optimization	An HN-I with built in dedicated trackers for PCIe peer-to-peer traffic.
HN-V	Home Node I/O + Distributed DVM Node	A device that includes an HN-I and DVM Node (DN).
SN-F	Subordinate Node	A memory controller consisting of a native CHI-C, CHI-D, or CHI-E SN interface.
SBSX	CHI to AXI or ACE-Lite bridge	A CHI to AXI or ACE-Lite bridge that allows an AXI or ACE-Lite memory controller to be connected to CMN-700.
CCG	CHI to CXS IssueB (CML_SMP/CXL) Bridge	CHI to CML_SMP/CXL bridge that enables CML, either for SMP connection, CXL Type3 device attachment, or memory expansion

The following table shows configurable options for mesh size, component counts, and top-level configuration (including associated parameters).

Table 2-3: Top-level configurable options: Mesh dimensions

Parameter	Description	Values (Default)	Comments
Mesh X dimension	Number of mesh columns	1-12	Mesh configurations that are not supported: • 1 x 1 • 1 x 2 • 2 x 1
Mesh Y dimension	Number of mesh rows	1-12	
MCSX count	Number of credited slices on an XP-XP mesh link in X dimension	0-4	This count is per link and can be different for each link.
MCSY count	Number of credited slices on an XP-XP mesh link in Y dimension	0-4	
DCS count	Number of credited slices on a device-XP link	0-4	
CCS count	Number of credited slices on a CAL-XP link	0-4	

Table 2-4: Top-level configurable options: global parameters

Parameter	Description	Values (Default)	Comments
RSVDC_METADATA_MODE_EN	Meta Data Preservation mode enable	0, 1 (False)	-
RSVDC_LOOPBACK_WIDTH	RSVDC LoopBack width	1, 2 (1)	Not supported in CMN-700
RSVDC_PBHA_MODE_EN	RSVDC PBHA mode enable	0, 1 (False)	-
RSVDC_PBHA_WIDTH	RSVDC PBHA width	2, 3, 4 (2)	-
RSVDC_STRONGNC_EN	StrongNC enable	0, 1 (False)	-
CHI_MPAM_ENABLE	MPAM feature enable	0, 1 (True)	-

Parameter	Description	Values (Default)	Comments
REQ_RSVDC_WIDTH	Width of RSVDC field in REQ flit	0, 4, 8, 16 (4)	-
REQ_ADDR_WIDTH	Width of ADDR field in REQ flit	44, 48, 52 (48)	REQ_ADDR_WIDTH must be set equal to or greater than PA_WIDTH. Address width of 52 is not supported in a CMN-700 system with RNF_CHIB_ESAM or RNF_CHIC_ESAM devices. For these configurations, CMN-700 supports a maximum REQ_ADDR_WIDTH value of 48.
PA_WIDTH	System Physical Address width	34, 44, 48, 52 (48)	-
DATACHECK_EN	Datacheck enable	0, 1 (False)	Data Check refers to data byte parity checking.
NUM_REMOTE_RNF	Number of RNFs for CML configurations on the all the remote chips combined.	0-384 (0)	-
FLIT_PAR_EN	Flit Parity enable	0, 1 (True)	-
RNSAM_NUM_HTG	Number of hashed regions supported by the RN SAM. This includes SCG, AXID based HNP hashed groups, non-architectural CPAG	0-32 (4)	-
RNSAM_NP2_EN	Enable non-power of two HNF hashing scheme	0, 1 (False)	-
RNSAM_HIER_HASH_EN	Enable Hierarchical hashing scheme	0, 1 (False)	-
RNSAM_AXID_HASH_EN	Enable AXID based power of two hashing scheme	0, 1 (False)	-
RNSAM_HTG_RCOMP_EN	Enable Range based address comparison for Hashed groups. Program start address and end address for each HTG	0, 1 (False)	-
RNSAM_HTG_RCOMP_LSB	Defines the minimum size of HTG when POR_RNSAM_HTG_RCOMP_EN_PARAM = 1, 16 value defines minimum size as 64KB and 26 value defines minimum size as 64MB	16-26 (26)	-
RNSAM_COMPACT_HN_TABLES_EN	Enable COMPACT HN Tables in the design	0, 1 (False)	-
RNSAM_NUM_ADD_HASHED_TGT	Number of additional hashed target ID's supported by the RN SAM, beyond the local HNF count	0, 2, 4, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96 (0)	-
RNSAM_NUM_NONHASH_REGION	Number of non-hashed regions supported by the RN SAM	4-64 (8)	-
RNSAM_NONHASH_RCOMP_EN	Enable Range based address comparison for non-hashed groups. Program start address and end address for each non-hashed groups	0, 1 (False)	-

Parameter	Description	Values (Default)	Comments
RNSAM_NONHASH_RCOMP_LSB	Defines the minimum size of non-hashed group when POR_RNSAM_NONHASH_RCOMP_EN_PARAM = 1, 16 value defines minimum size as 64KB and 26 value defines minimum size as 64MB	16-26 (26)	-
RNSAM_NUM_CPA_GRP	Number of Architectural CPA groups	0-16 (5)	-
RNSAM_NUM_QOS_REGIONS	Number of memory regions for QoS override	0, 2, 4, 6, 8 (0)	-
RNSAM_PREFETCH_EN	Enables HNF->MC SAM in the RNSAM which can be used for prefetch type transactions. RNSAM generates a valid SN targetID when this parameter is enabled	0, 1 (True)	-
RNSAM_PFTGT_NUM_SCG	Number of SCG's supported for prefetch transactions by the RNSAM	0-8 (4)	-
RNSAM_PFTGT_NUM_NONHASH_PSCG	Number of prefetch non-hashed regions supported per System Cache Group by the RNSAM	0-64 (0)	-
RNSAM_PFTGT_NUM_HTG_PSCG	Number of prefetch HTG regions supported per System Cache Group by the RNSAM	0-8 (0)	-
RNSAM_FLEX_TGTID_EN	Enables Flexible target ID table base indexes for HNF and CCG target ID's. To support backward compatible, set this parameter to zero	0, 1 (False)	-
RNSAM_CUSTOM_REGS	Number of customer specific registers for customer implemented logic	0-8 (0)	-
HNSAM_NUM_NONHASH	Number of non-hashed regions supported by the HNSAM	0-64 (2)	-
HNSAM_NUM_HTG	Number of HTG regions supported by the HNSAM	0-16 (0)	-
HNSAM_RCOMP_EN	Enable Range based address comparison for HNSAM HTG/Nonhashed groups. Program start address and end address.	0, 1 (False)	-
HNSAM_RCOMP_LSB	Defines the minimum size of HTG when POR_HNSAM_RCOMP_EN_PARAM = 1, 20 value defines minimum size as 1MB and 26 value defines minimum size as 64MB	20-26 (26)	-
HNSAM_CUSTOM_REGS	Number of customer specific registers for customer implemented logic	0-8 (0)	-
EN_2X_DAT_VC	Enables 2x DAT VC	0, 1 (False)	-
EN_2X_RSP_VC	Enables 2x RSP VC	0, 1 (False)	-
EN_2X_REQ_VC	Enables 2x REQ VC	0, 1 (False)	-
EN_2X_SNP_VC	Enables 2x SNP VC	0, 1 (False)	-
MESH2X_DEF_SEL	Selects the default ping-pong scheme for TGTID Lookup in 2xMESH. 0 -> Default Ping-Pong scheme based on Even/Odd XID, 1 -> Default Ping-Pong scheme based on Even/Odd YID	0, 1 (0)	-
CHI_MTE_ENABLE	Memory Tagging feature enable	0, 1 (True)	-
XY_OVERRIDE_CNT	Number of Src-Tgt pairs whose XY route path can be overriden	0, 2, 4, 8, 16 (0)	-

Parameter	Description	Values (Default)	Comments
MXP_MULTIPLE_DTM_EN	Multiple DTMs feature enable. This is used if number of device ports on the XP is > 2	0, 1 (False)	-
RXBUF_NUM_ENTRIES_MCS	Number of entries in the RX Buffer at upload interface of MCSX and MCSY	2-4 (2)	The minimum value of 2 corresponds to a credit return latency of one cycle in the SMXP and one cycle in the MCSX/MCSY.
MCS_PUB_RSL_EN	Enable Register Slice on PUB outputs in MCS	0, 1 (False)	-
MXP_DAT_WH_ROUTE_EN	Enable Worm Hole Routing on DAT Channel in MXP	0, 1 (False)	Not supported in CMN-700
MXP_AXU_EN	Enable AXU interface on all MXP	0, 1 (False)	-

Table 2-5: Top-level configurable options: various resources

Feature	Parameter	Description	Values (default)	Comments
Clock resources	Number of clock inputs	The number of clock inputs in a synchronous or asynchronous mesh	1, 4 (1)	-
Processor resources	Number of RN-Fs	The number of RN-Fs in the system.	1-256 without CAL 2-256 with CAL	RN-Fs can be one of the following four types: • RNF_CHIB_ESAM • RNF_CHIC_ESAM • RNF_CHID_ESAM • RNF_CHIE_ESAM
I/O resources	Number of RN-Is	The number of RN-I instances in the system	0-90	At least one RN-I or RN-D must be present. The total count of RN-Is and RN-Ds must not exceed 90.
	Number of RN-Ds	The number of RN-D instances in the system	0-40	
	Number of HN-Is	The number of HN-I instances in the system.	0-32 2-32 with CAL	This count includes HN-T, HN-P and the HN-D which is always present
Debug resources	Number of HN-T (HNI + DTC)	The total number of Debug Trace Controller domains.	1-4	The number of DTCs must not exceed the number of HN-Is.
System cache	Number of HN-Fs	The total number of HN-F instances in the system.	1-64 without CAL 2-128 with CAL	For more details, refer to 5. SLC memory system on page 1157.

Feature	Parameter	Description	Values (default)	Comments
Memory resources	Number of SN-Fs	The number of SN-Fs (CHI interfaces)	0-64 without CAL 2-128 with CAL	At least one SN-F or SBSX must be present. The total count of SN-Fs and SBSXs must not exceed 128. CMN-700 supports: • A configurable number of CHI-C thru CHI-E memory ports (SNF and SBSX) with CAL to support 6x HBM3 stacks and 4x DDR5 DIMM channels
	Number of SBSXs	The number of SBSX instances (AXI interfaces)		
	Number of CCGs	The number of CCG instances	0-32	-

2.5.3 Device placement and configuration

When the devices are enumerated and the mesh dimensions are determined, the placement of each device or node in the mesh must be specified.

While there are no constraints on the mesh location of a device, floorplanning and performance constraints drive the optimal device placement. This detail is outside the scope of this document.

The following table shows the options that you can use to configure individual CMN-700 devices.


Note

When CAL is present, all the devices that are connected to it must be configured identically.

Table 2-6: CHI device configurable options: RNF-port, SNF-port

Parameter	Description	Values (Default)	Comments
POISON	Data poison enable (RN-F port only)	0, 1 (True)	-
DATACHECK	Data Check enable (RN-F port only)	0, 1 (False)	End-to-end data byte parity enable
RXBUF_NUM	Number of receive flit buffers inside CMN-700 on this port.	2-4 (3)	To achieve full bandwidth operation, this number must equal the CHI credit return latency (in cycles) for flit transfers from RN-F or SNF to the interconnect. The minimum value of 2 corresponds to a credit return latency of one cycle in the interconnect and one cycle in the RN-F or SN-F.

Table 2-7: CHI device configurable options: RN-I, RN-D

Parameter	Description	Values (Default)	Comments
AXDATA_WIDTH	Data width on AXI or ACE-Lite interface	128, 256, 512 (128)	-
NUM_WR_REQ	Number of Write Request Tracker entries	4, 16, 24, 32, 64, 128 (32)	If NUM_WR_REQ = 128, the WDBs are implemented using RAM only. Otherwise, flop-based implementation is used
NUM_ATOMIC_BUF	Depth of Atomic data buffers	2, 4, 8, 16, 32 (2)	-
NUM_RD_REQ	Number of Read Request Tracker entries	4, 32, 64, 96, 128, 256, 512 (32)	If NUM_RD_BUF is 128, 256, or 512, NUM_RD_REQ must be the same value. The number of tracker entries must be the same or larger than data buffer entries. NUM_RD_REQ ≥ NUM_RD_BUF.
NUM_RD_REQ_MAX_PER_SLICE	Number of Read Request Tracker entries per slice	64, 128 (64)	When NUM_RD_REQ is 128 or greater, this is the maximum number of tracker entries per slice. The number of read slices is determined by NUM_RD_REQ divided by NUM_RD_REQ_MAX_PER_SLICE.
NUM_RD_BUF	Number of Read Data Buffers.	4, 8, 16, 24, 32, 64, 96, 128, 256, 512 (24)	This value must be 256 when NUM_RD_REQ is 256 or must be 512 when NUM_RD_REQ is 512 and should be less than or equal to NUM_RD_REQ for all other cases. NUM_RD_BUF ≥ 128 instantiates RAM for data buffer.
NUM_PREALLOC_RD_BUF	Number of Pre-allocated Read Data Buffers.	4, 8, 16, 32 (8)	This value must be greater than NUM_RD_BUF.
AXDATAPOISON_EN	Data Poison Enable on AXI or ACE-Lite interface	0, 1 (0)	-
AXLOOPBACK_EN	2-bit Loopback Enable on AXI or ACE-Lite interface	0, 1 (0)	Enabling this parameter permits the RN-I and RN-D to store information for faster transactions.
AXMPAM_EN	Enables MPAM feature on AXI or ACE-Lite interfaces	0, 1 (True)	Enables the *Memory Partioning and Monitioring* (MPAM) feature for RN-I and RN-D.
FORCE_RDB_PREALLOC	Force read data buffer pre-allocation	0, 1 (0)	-
AXID_WIDTH	ID width for subordinate ports	11, 16, 24, 32 (11)	-
LEGACY_DECOUP_RD	Enable legacy decoupled read logic	0, 1 (1)	This parameter only affects decoupled mode where NUM_RD_BUF < NUM_RD_REQ, otherwise has no effect.

Table 2-8: CHI device configurable options: HN-F

Parameter	Description	Values (Default)	Comments
SLC_SIZE	Size of system cache per HN-F node.	-8, -2, -1, 0, 1, 2, 3 (2)	Valid SLC_SIZE, SLC_NUM_WAYS combinations are **0K** (SLC_SIZE= -8, NUM_WAYS = 16) **128K** (SLC_SIZE= -2, NUM_WAYS = 16) **256K** (SLC_SIZE= -1, NUM_WAYS = 16) **384K** (SLC_SIZE= 0, NUM_WAYS = 12) **512K** (SLC_SIZE= 0, NUM_WAYS = 16) **1M** (SLC_SIZE= 1, NUM_WAYS = 16) **1.5M** (SLC_SIZE= 2, NUM_WAYS = 12) **2M** (SLC_SIZE= 2, NUM_WAYS = 16) **3M** (SLC_SIZE= 3, NUM_WAYS = 12) **4M** (SLC_SIZE= 3, NUM_WAYS = 16)
SF_SIZE	Size of SF tag RAM	0, 1, 2, 3, 4 (3)	Size of the SF tag RAM is chosen based on `SF_NUM_WAYS` as: SF_SIZE = (32K * (2^SF_SIZE_PARAM) * SF_NUM_WAYS) • `SF_NUM_WAYS` = 16: 512K, 1M, 2M, 4M, 8M • `SF_NUM_WAYS` = 32: 1M, 2M, 4M, 8M, 16M
SLC_NUM_WAYS	Number of ways in the system level cache, set to 12 for 384K, 1.5M, 3M SLC size	12, 16 (16)	Valid SLC_SIZE, SLC_NUM_WAYS combinations are **0K** (SLC_SIZE= -8, NUM_WAYS = 16) **128K** (SLC_SIZE= -2, NUM_WAYS = 16) **256K** (SLC_SIZE= -1, NUM_WAYS = 16) **384K** (SLC_SIZE= 0, NUM_WAYS = 12) **512K** (SLC_SIZE= 0, NUM_WAYS = 16) **1M** (SLC_SIZE= 1, NUM_WAYS = 16) **1.5M** (SLC_SIZE= 2, NUM_WAYS = 12) **2M** (SLC_SIZE= 2, NUM_WAYS = 16) **3M** (SLC_SIZE= 3, NUM_WAYS = 12) **4M** (SLC_SIZE= 3, NUM_WAYS = 16)
SF_NUM_WAYS	Number of ways in Snoop Filter cache	16, 20, 24, 28, 32 (16)	-
SLC_TAG_RAM_LATENCY	Latency of system cache and snoop filter tag RAM	1, 2, 3 (2)	Valid Tag:Data combinations are 1:2, 2:2, 3:3
SLC_DATA_RAM_LATENCY	Latency of system cache data RAM	2, 3 (2)	Valid Tag:Data combinations are 1:2, 2:2, 3:3
NUM_ENTRIES_POCQ	Number of entries in the POCQ tracker	16, 24, 32, 48, 64, 80, 96, 128 (32)	> 64 depths may have frequency implications *Point-of -Coherency Queue* (POCQ)
SF_RN_ADD_VECTOR_WIDTH	Number of addiitonal bits in the Snoop Filter to track the RN-Fs	0-64 (0)	-
SF_MAX_RNF_PER_CLUSTER	Maximum number of RN-Fs per cluster as represented in the SF's RN-F vector	1, 2, 4, 8 (1)	-
MPAM_NS_PARTID_MAX	Maximum value of non-secure MPAM partitions	1, 2, 4, 8, 16, 32, 64, 128, 256, 512 (64)	-

Parameter	Description	Values (Default)	Comments
MPAM_S_PARTID_MAX	Maximum value of secure MPAM partitions	1, 2, 4, 8, 16, 32, 64, 128, 256, 512 (16)	-
MPAM_NS_PMG_MAX	Maximum value of non-secure MPAM PMGs	1, 2 (2)	-
MPAM_S_PMG_MAX	Maximum value of secure MPAM PMGs	1, 2 (2)	-
MPAM_NUM_CSUMON	Number of CSU monitering counters	1, 2, 4, 8, 16 (4)	-

Table 2-9: CHI device configurable options: HN-I, HN-D, HN-T, HN-V

Parameter	Description	Values (Default)	Comments
AXDATA_WIDTH	Data width on AXI or ACE-Lite interface	128, 256, 512 (128)	-
NUM_RRT_REQS	Depth of Request Receive Tracker (RRT)	8, 16, 32, 60, 124 (8)	-
NUM_AXI_REQS	Number of request tracker entries	8, 32, 64, 128, 256 (32)	-
AXDATAPOISON_EN	Data poison enable on AXI or ACE-Lite interface	0, 1 (True)	-
AXMPAM_EN	Enables MPAM feature on AXI or ACE-Lite interfaces	0, 1 (True)	-

Table 2-10: CHI device configurable options: HN-P

Parameter	Description	Values (Default)	Comments
AXDATA_WIDTH	Data width on AXI or ACE-Lite interface	128, 256, 512 (128)	-
NUM_RRT_REQS	Depth of Request Receive Tracker (RRT)	8, 16, 32, 60, 124 (8)	-
NUM_AXI_REQS	Number of request tracker entries	8, 32, 64, 128, 256 (32)	-
AXDATAPOISON_EN	Data poison enable on AXI or ACE-Lite interface	0, 1 (True)	-
AXMPAM_EN	Enables MPAM feature on AXI or ACE-Lite interfaces	0, 1 (True)	-
WR_NUM_RRT_REQS	Depth of P2P Write Slice Request Receive Tracker (RRT)	16, 32, 64 (16)	-
WR_NUM_AXI_REQS	Depth of P2P Write Slice OT AXI Request Tracker	32, 64 (32)	-
RD_NUM_AXI_REQS	Depth of P2P Read Slice OT AXI Request Tracker	64, 128, 256 (64)	-

Table 2-11: CHI device configurable options: CCG

Parameter	Description	Values (Default)	Comments
PCIE_ENABLE	When set, enables PCIe traffic through this CCG	0, 1 (True)	-
HA_PRESENT	When set,instantiates HA in this CCG	0, 1 (True)	-
RA_NUM_REQS	Depth of Request Tracker	64, 128, 256 (256)	-
RA_NUM_RDBUF	Depth of Read Data Buffer.	16, 24, 32 (16)	-
RA_NUM_WRBUF	Depth of Write Data Buffer.	16, 24, 32, 64 (24)	-

Parameter	Description	Values (Default)	Comments
RA_NUM_SNPREQS	Depth of Snoop Tracker	8, 64, 128, 256 (128)	-
RA_NUM_SNPBUF	Depth of Snoop Data Buffer.	4, 16, 24, 32, 64 (32)	-
HA_NUM_REQS	Depth of request tracker.	128, 192, 256 (192)	-
HA_NUM_WRBUF	Depth of Write Data Buffer.	64, 96, 128 (64)	-
HA_NUM_SNPREQS	Depth of Snoop Tracker. This indicates the number of outstanding snoop request HA can have on CCIX.	96, 128, 256 (96)	-
HA_NUM_SNPBUF	Depth of Snoop Data Buffer.	16, 24, 32 (24)	-
HA_SSB_DEPTH	This depth indicates the maximum number of remote snoop requests from the local chip that can be sunk at this HA.	96, 128, 256, 512 (96)	To avoid deadlock, this queue must sink in all snoops coming to that HA. HN-Fs must ensure that no more than this number is sent to the HA. This value must be at least as large as CCG_HA_NUM_SNPREQS_PARAM
HA_REQ_PASS_BUFF_DEPTH	Depth of req channel passive buffer.	64, 128, 256, 512 (256)	-
HA_DAT_PASS_BUFF_DEPTH	Depth of data channel passive buffer.	64, 128, 256, 512 (256)	-
NUM_WR_REQ	Number of Write Request Tracker entries	4, 16, 24, 32, 64, 128 (32)	-
NUM_RD_REQ	Number of Read Request Tracker entries	4, 32, 64, 96, 128, 256, 512 (32)	If NUM_RD_BUF is 128, 256 or 512, NUM_RD_REQ must be the same value. The number of tracker entries must be the same or larger than data buffer entries. NUM_RD_REQ ≥ NUM_RD_BUF.
NUM_RD_BUF	Number of Read Data Buffers.	4, 32, 64, 96, 128, 256, 512 (32)	This value must be 256 when NUM_RD_REQ is 256 and this value must be 512 when NUM_RD_REQ is 512. This value should be less than or equal to NUM_RD_REQ for all other cases. NUM_RD_BUF > 128 instantiates RAM for data buffer.
NUM_RD_REQ_MAX_PER_SLICE	Number of Read Request Tracker entries per slice	64, 128 (64)	When NUM_RD_REQ is 128 or greater, this is the maximum number of tracker entries per slice. The number of read slices is determined by NUM_RD_REQ divided by NUM_RD_REQ_MAX_PER_SLICE.

Parameter	Description	Values (Default)	Comments
NUM_ATOMIC_BUF	Depth of Atomic data buffers	2, 4, 8, 16, 32 (8)	-
LEGACY_DECOUP_RD	Enable legacy decoupled read logic	0, 1 (1)	-
RX_STL_BUFFER_DEPTH	Depth of CCL stalling channel RX buffer for CXS RSP with data messages	32, 64, 128, 256 (256)	-

Table 2-12: CHI device configurable options: SBSX

Parameter	Description	Values (Default)	Comments
NUM_DART	Number of DART tracker entries	64, 128 (64)	-
NUM_WR_BUF	Number of write buffers	8, 16 (8)	-
AXDATA_WIDTH	Data width on AXI or ACE-Lite interface	128, 256 (128)	-
AXDATAPOISON_EN	Data Poison enable on AXI or ACE-Lite interface	0, 1 (True)	-
AXMPAM_EN	Enables MPAM feature on AXI or ACE-Lite interfaces	0, 1 (True)	-
CMO_ON_AW	Enables Write Channel CMOs on AXI or ACE-Lite interface.	0, 1 (False)	If Enabled, CMOs are sent only on AW channel.

Table 2-13: CHI device configurable options: CDB SN-F

Parameter	Description	Values (Default)	Comments
REQLCRD_MAX_COUNT_DEV	Max number of Link Credits on REQ channel given to DEV	1-15 (4)	-
RSPLCRD_MAX_COUNT_DEV	Max number of Link Credits on RSP channel given to DEV	1-15 (4)	-
DATLCRD_MAX_COUNT_DEV	Max number of Link Credits on DAT channel given to DEV	1-15 (4)	-
SNPLCRD_MAX_COUNT_DEV	Max number of Link Credits on SNP channel given to DEV	1-15 (4)	-

Table 2-14: CHI device configurable options: CDB RN-F

Parameter	Description	Values (Default)	Comments
REQLCRD_MAX_COUNT_DEV	Max number of Link Credits on REQ channel given to DEV	1-15 (4)	-
RSPLCRD_MAX_COUNT_DEV	Max number of Link Credits on RSP channel given to DEV	1-15 (4)	-
DATLCRD_MAX_COUNT_DEV	Max number of Link Credits on DAT channel given to DEV	1-15 (4)	-
SNPLCRD_MAX_COUNT_DEV	Max number of Link Credits on SNP channel given to DEV	1-15 (4)	-

Table 2-15: CHI device configurable options: ADB RN-D

Parameter	Description	Values (Default)	Comments
AW_FIFO_DEPTH	AW channel FIFO depth depth	4, 6, 8, 10 (8)	-
W_FIFO_DEPTH	W channel FIFO depth depth	4, 6, 8, 10 (8)	-
AR_FIFO_DEPTH	AR channel FIFO depth depth	4, 6, 8, 10 (8)	-
CR_FIFO_DEPTH	CR channel FIFO depth depth	4, 6, 8, 10 (8)	-
B_FIFO_DEPTH	B channel FIFO depth depth	4, 6, 8, 10 (8)	-

Parameter	Description	Values (Default)	Comments
R_FIFO_DEPTH	R channel FIFO depth depth	4, 6, 8, 10 (8)	-
AC_FIFO_DEPTH	AC channel FIFO depth depth	4, 6, 8, 10 (8)	-

Table 2-16: CHI device configurable options: ADB RN-I

Parameter	Description	Values (Default)	Comments
AW_FIFO_DEPTH	AW channel FIFO depth depth	4, 6, 8, 10 (8)	-
W_FIFO_DEPTH	W channel FIFO depth depth	4, 6, 8, 10 (8)	-
AR_FIFO_DEPTH	AR channel FIFO depth depth	4, 6, 8, 10 (8)	-
B_FIFO_DEPTH	B channel FIFO depth depth	4, 6, 8, 10 (8)	-
R_FIFO_DEPTH	R channel FIFO depth depth	4, 6, 8, 10 (8)	-

Table 2-17: CHI device configurable options: ADB HN-I/D/T/V

Parameter	Description	Values (Default)	Comments
AW_FIFO_DEPTH	AW channel FIFO depth depth	4, 6, 8, 10 (8)	-
W_FIFO_DEPTH	W channel FIFO depth depth	4, 6, 8, 10 (8)	-
AR_FIFO_DEPTH	AR channel FIFO depth depth	4, 6, 8, 10 (8)	-
B_FIFO_DEPTH	B channel FIFO depth depth	4, 6, 8, 10 (8)	-
R_FIFO_DEPTH	R channel FIFO depth depth	4, 6, 8, 10 (8)	-

Table 2-18: CHI device configurable options: ADB HN-P

Parameter	Description	Values (Default)	Comments
AW_FIFO_DEPTH	AW channel FIFO depth depth	4, 6, 8, 10 (8)	-
W_FIFO_DEPTH	W channel FIFO depth depth	4, 6, 8, 10 (8)	-
AR_FIFO_DEPTH	AR channel FIFO depth depth	4, 6, 8, 10 (8)	-
B_FIFO_DEPTH	B channel FIFO depth depth	4, 6, 8, 10 (8)	-
R_FIFO_DEPTH	R channel FIFO depth depth	4, 6, 8, 10 (8)	-

Table 2-19: CHI device configurable options: ADB SBSX

Parameter	Description	Values (Default)	Comments
AW_FIFO_DEPTH	AW channel FIFO depth depth	4, 6, 8, 10 (8)	-
W_FIFO_DEPTH	W channel FIFO depth depth	4, 6, 8, 10 (8)	-
AR_FIFO_DEPTH	AR channel FIFO depth depth	4, 6, 8, 10 (8)	-
B_FIFO_DEPTH	B channel FIFO depth depth	4, 6, 8, 10 (8)	-
R_FIFO_DEPTH	R channel FIFO depth depth	4, 6, 8, 10 (8)	-

Table 2-20: CHI device configurable options: CXSDB CCG

Parameter	Description	Values (Default)	Comments
CXS_CRD_MAX_COUNT_DEV	Max number of Link Credits given to DEV	1-15 (4)	-
CXS_FIFO_DEPTH_DEV	CXS DEV->ICN FIFO depth	4, 6, 8, 10 (8)	-
CXS_CRD_MAX_COUNT_ICN	Max number of Link Credits given to ICN	1-15 (4)	-
CXS_FIFO_DEPTH_ICN	CXS ICN->DEV FIFO depth	4, 6, 8, 10 (8)	-

2.6 Test features

The CMN-700 product includes several test features.

See the *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual* for information about the test features.

2.7 Product documentation and design flow

The CMN-700 product manuals support the design flow process.

Documentation

The CMN-700 product is supported by the following documentation:

Technical Reference Manual

The *Technical Reference Manual* (TRM) describes the functionality, and how functional options affect the behavior of CMN-700. It is required at all stages of the design flow. The choices that you make in the design flow can mean that some behavior that is described in the TRM is not relevant. If you are programming the CMN-700 product, contact:

- The implementer to determine:
  - The build configuration of the implementation
  - What integration, if any, was performed before implementing the CMN-700 product
- The integrator to determine the pin configuration of the device that you are using

Configuration and Integration Manual

The *Configuration and Integration Manual* (CIM) describes how to integrate the CMN-700 product into an SoC. It includes a description of the pins that the integrator must tie off to configure the macrocell for the required integration. The CIM also 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 on 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.

User Guide

The Socrates™ *User Guide* describes how to use Socrates™ to configure and integrate a custom mesh interconnect.


Note

The *Arm® Socrates™ User Guide* is part of the Socrates™ product download bundle.

Design flow

CMN-700 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 process includes integrating RAMs into the design.

Integration

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

Programming

Programming is the last process. The system programmer develops the software that is required to configure and initialize the CMN-700 product, and tests the required application software.

Each process can:

- Be performed by a different party
- Include implementation and integration choices that affect the behavior and features of the CMN-700 product

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 CMN-700 product 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 CMN-700 product by programming particular values into registers. The register configuration affects the behavior of the CMN-700 product.


Note

This book refers to **IMPLEMENTATION DEFINED** features that are applicable to build configuration options. A reference to a feature that is included means that the appropriate build and pin configuration options are selected. A reference to an enabled feature means a feature that the software has configured.

2.8 Product revisions

The following describes the differences in functionality between successive product revisions of the CMN-700 product.

r0p0

First release

r1p0

Second release:

- CXLv2.0 host side support of CXL.mem and CXL.io protocols for Type-3 memory expansion devices
- 32 CCG or CXG gateway nodes
- Non power-of-2 hashing of HN-Fs with 2N * {1, 3, or 5} up to 64 HN-Fs or 128 HN-Fs with CAL

r2p0

Third release:

- Remote PCIe Streaming Support
- 1.5MB SLC support
- 90 RN-I support
- 128 SN-F/ SBSX support
- AXID based for port aggregation across chip
- RNSAM support for 4 chip flat hashing configuration

r3p0

Fourth release:

- AXU port on all MXPs
- 512 RNI requests support
- 16 bit REQ RSVDC support
- Configurable write cancel threshold in RN-I and RN-D
- Remote DVM Sync Collapsing
- CPAG MOD-3 Hashing
- PCIe write streaming improvements

r3p2

Fifth release:

- Performance optimization guideline improvements for RN-I and RN-D
- HN-P and HN-I AxID Encoding improvements

3. Functional description

This chapter describes the functionality that is achieved when you design and configure the CMN-700 interconnect and its components.

3.1 Components and structural configuration

When configuring CMN-700, the components you select depend on the required functionality. The selected components, topology, and other configurable options affect the overall structure of CMN-700 and the associated behavior.

The following contains the following subsections:

- 3.1.1 Components on page 40 describes the individual interconnect components that form CMN-700.
- 3.1.2 System configurations on page 56 describes the possible system-level interconnect topologies, including example interconnect configurations.
- 3.1.3 CML system configurations on page 62 describes the possible CML system topologies, including example configurations and information about specific CML options.
- 3.1.4 Structural configuration and considerations on page 68 describes optional configurations that impact the structure and behavior of CMN-700.

3.1.1 Components

CMN-700 is made up of various types of devices, including router modules, CHI nodes, and bridges. The components that you require depend on the requirements of your system and some are optional or only used if certain requirements are met.

CMN-700 can be integrated into a complete SoC system that includes devices that are not described in this section.

3.1.1.1 Crosspoint

The *Crosspoint* (XP) is a switch or router logic module. It is the fundamental component building block of the CMN-700 transport mechanism.


Note

The terms *Crosspoint (XP)* and *Mesh Crosspoint (MXP)* are used interchangeably throughout the

The CMN-700 mesh interconnect is built using a set of XP modules. The XP modules are arranged in a two-dimensional rectangular mesh topology. Each XP can connect to up to four neighboring

XPs using mesh ports that are shown with the dashed lines in the following figure. Each XP also has two device ports for connecting devices, P0 and P1.

Depending on the configuration, the XP can have a maximum of four device ports for mesh configurations.

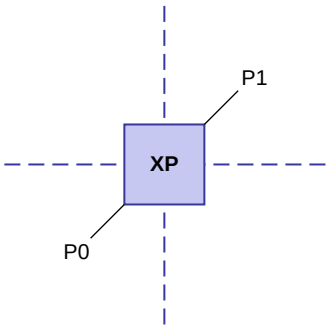
***MXPs* with 4 mesh ports**
Can support up to 2 device ports

***MXPs* on the edge of the mesh with 3 mesh ports**
Can support up to 3 device ports

***MXPs* on the corners of the mesh with 2 mesh ports**
Can support up to 4 device ports

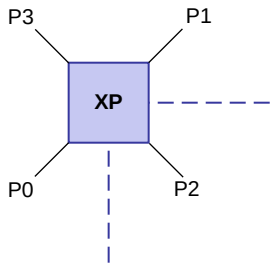
See 3.1.4.3 Dual CHI channel selection registers on page 71.

Figure 3-1: Crosspoint (XP)



MXPs on the corners of the mesh with 2 mesh ports can support up to 4 device ports, as shown in the following figure.

Figure 3-2: Crosspoint structure in mesh configuration with extra device ports

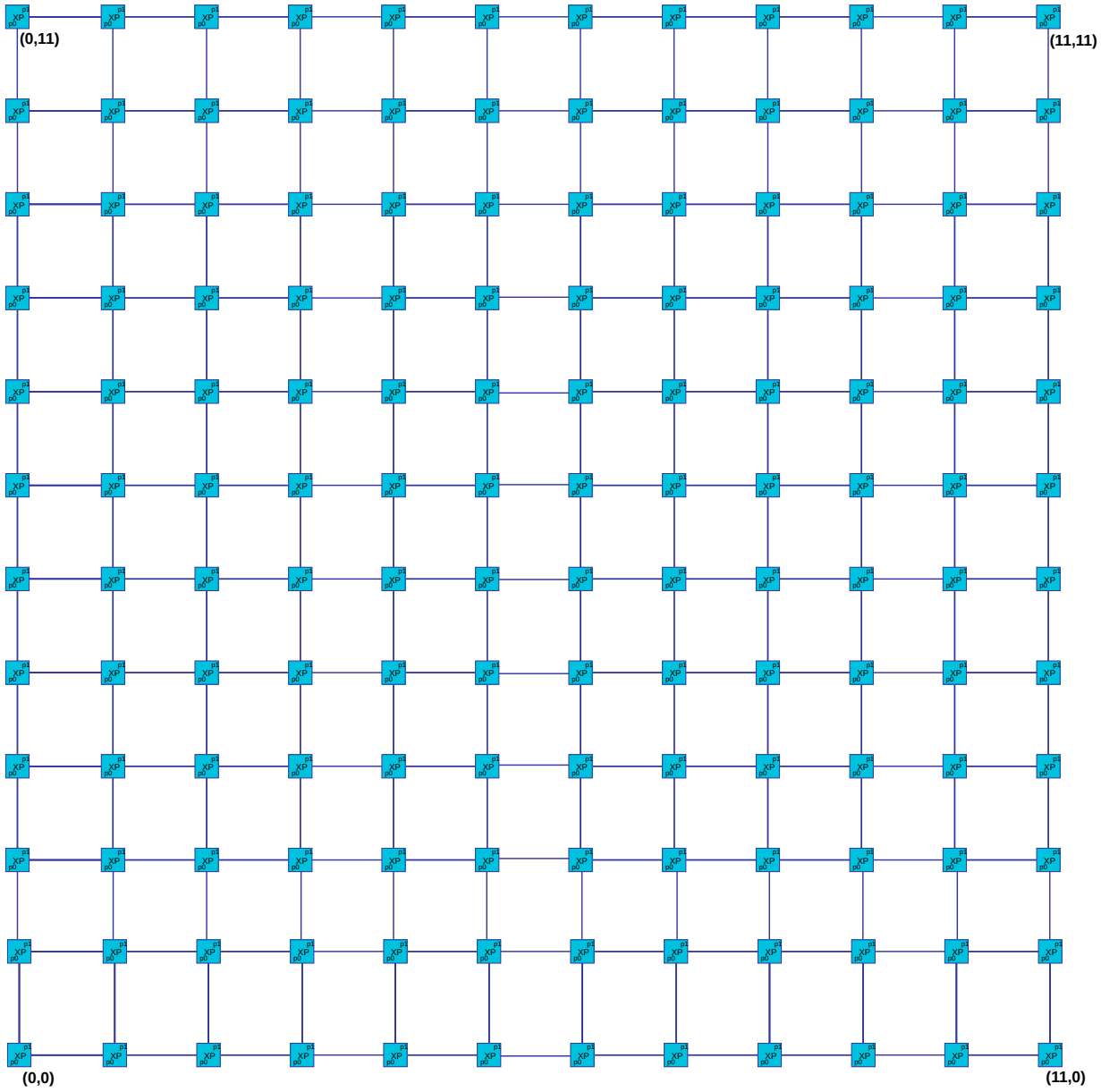


Each XP supports four CHI channels for transporting flits across the mesh from a source device to a destination or target device:

- *Request* (REQ)
- *Response* (RSP)
- *Snoop* (SNP)
- *Data* (DAT)

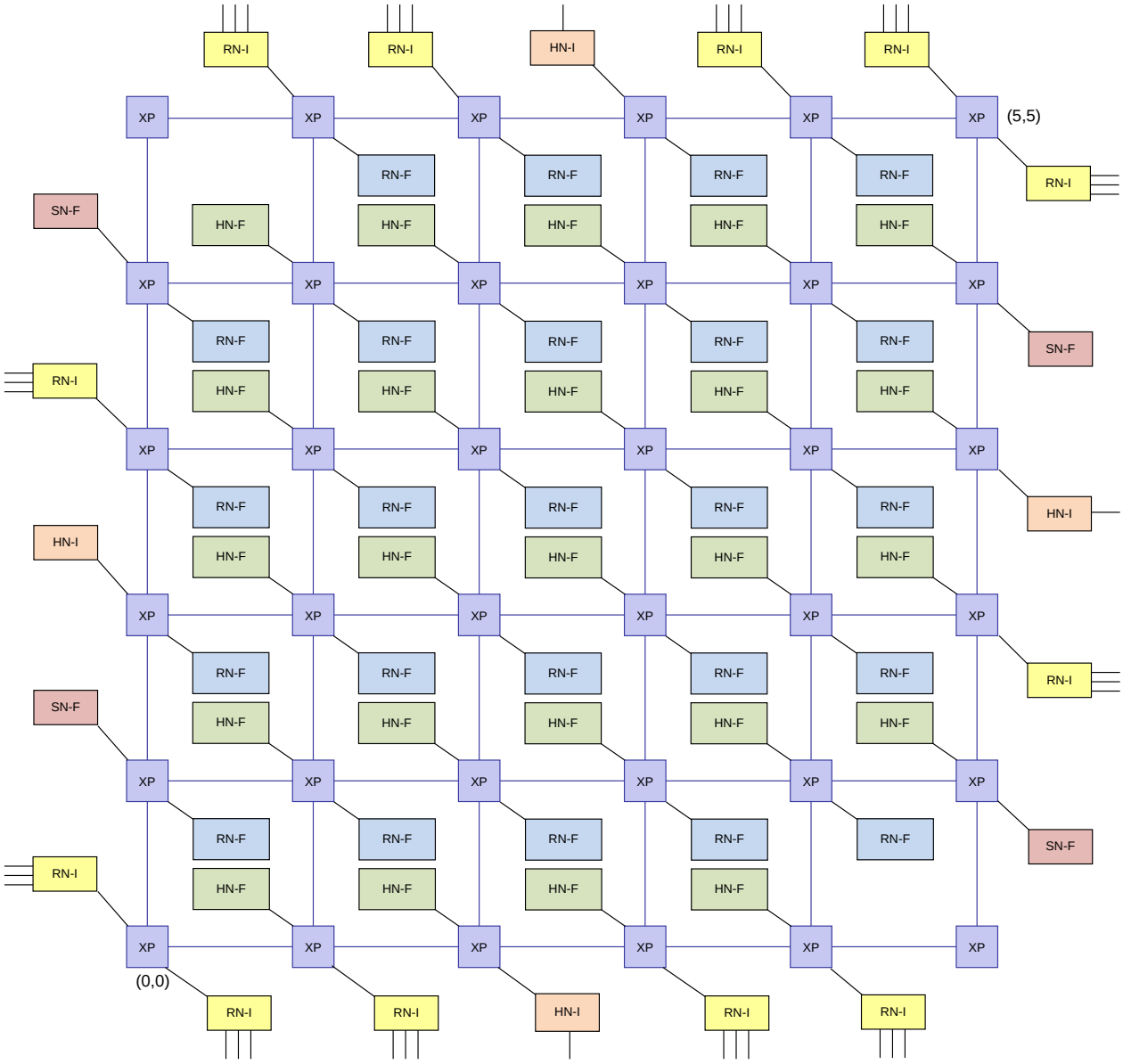
The maximum size for the CMN-700 mesh is 144 XPs arranged in a 12 × 12 grid. Each XP in the grid is referenced using an (X,Y) coordinate system. (0,0) represents the bottom-left corner, and a maximum coordinate of (11,11) represents the upper-right corner. The following figure shows the maximum 12 × 12 mesh configuration with some (X,Y) coordinate values.

Figure 3-3: 12 × 12 maximum mesh configuration



The following figure shows an example 6 × 6 mesh configuration, with devices attached to XP ports.

Figure 3-4: Example 6 × 6 mesh configuration



Note

The X and Y coordinates of an XP are also known as the XID and YID respectively.

3.1.1.2 I/O coherent Request Node

The *I/O-coherent Request Node* (RN-I) connects I/O-coherent AMBA managers to the rest of the CMN-700 system.

An RN-I bridge includes three ACE-Lite or ACE-Lite-with-DVM subordinate ports.

The RN-I bridge can act as a proxy only for managers that do not contain hardware-coherent caches. There is no capability to issue snoop transactions to RN-Is.

3.1.1.3 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:

System Level Cache

The *System Level Cache* (SLC) is a last-level cache. The SLC allocation policy is exclusive for data lines, except where sharing patterns are detected and pseudo-inclusive for code lines, as indicated by the RN-Fs. All code lines can be allocated into the SLC on the initial request. When MTE is enabled, SLC stores data and tags.

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. Ordering includes serialization of multiple outstanding requests and actions to the same line, and request ordering as required by the RN-F.

Snoop Filter

The *Snoop Filter* (SF) tracks cachelines that are present in the RN-Fs. It reduces snoop traffic in the system by favoring directed snoops over snoop broadcasts when possible. This approach substantially reduces the snoop response traffic that might otherwise be required.

Each HN-F in the system is configured to manage a specific portion of the overall address space. This specific portion of address space will be dictated by the SAM mappings, see 4.4.3 RN SAM and HN-F SAM programming on page 1121 for more information.

The entire DRAM space is managed through the combination of all HN-Fs in the system.

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

3.1.1.4 I/O coherent Home Node

The *I/O coherent Home Node* (HN-I) is a Home Node for all CHI transactions targeting AMBA subordinate devices.

The HN-I acts as a proxy for all the RNs of CMN-700, converting CHI transactions to ACE5-Lite transactions. The HN-I includes support for the correct ordering of Arm device types.

The HN-I does not support caching of any data read from or written to the downstream ACE5-Lite I/O subordinate subsystem. Any cacheable request that is sent to the HN-I does not result in any snoops being sent to RN-Fs in the system. Instead, the request is converted to the appropriate ACE5-Lite read or write command and sent to the downstream ACE5-Lite subsystem.

If an RN-F caches data that is read from or written to the downstream ACE5-Lite I/O subordinate subsystem, coherency is not maintained. Any subsequent access to that data reads from or writes to the ACE5-Lite I/O subordinate subsystem directly, ignoring the cached data.

3.1.1.5 I/O coherent Home Node with PCIe optimization

The *I/O coherent Home Node with PCIe optimization* (HN-P) is a device that includes the HN-I functionality and dedicated trackers for PCIe peer-to-peer traffic.

HN-P can only be used to connect to PCIe subordinates.

3.1.1.6 SBSX

The *AMBA 5 CHI to ACE5-Lite bridge* (SBSX) functions as a CHI SN-F and enables an ACE5-Lite subordinate device to be used in a CMN-700 system.

3.1.1.7 CML

A *Coherent Multichip Link* (CML) device enables multi-chip communication in CMN-700.

A given multi-chip link can be used for:

- *SMP* (CML_SMP) connection
- *CXL* device attachment

A CML device (CCG) can be configured to be used for CML_SMP connection or CXL device attachment.

For *CML_SMP* connection an I/O coherent Requesting Node (RN-I) must be included in the *CCG* block to speed up PCIe traffic targeting remote memory.

3.1.1.8 Configuration node

The *configuration node* (CFG) is co-located with the HN-D node and handles various CMN-700 configuration, control, and monitoring features.

The CFG carries out the following functions:

- Configuration accesses
- Error reporting and signaling
- Interrupt generation

The CFG includes the following elements:

- Ports to collect error signals from CHI components within CMN-700
- A configuration bus which connects to all nodes to handle internal configuration register reads and writes
- A dedicated APB interface for configuration accesses

The CFG does not have a dedicated CHI port. It shares a device port with the HN-D node in the mesh.

3.1.1.9 Power/Clock Control Block

The *Power/Clock Control Block* (PCCB), co-located with the HN-D node, provides separate communication channels. These channels 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 CMN-700 components, in the following manner:

1. The PCCB receives transaction activity indicators from other relevant CMN-700 components and conveys that information to the external power and clock control units.
2. The PCCB receives power or clock control management requests from the external power or clock control units. Where applicable, it conveys that request to the relevant CMN-700 components.
3. The PCCB waits for the appropriate responses from the relevant CMN-700 components, and conveys an aggregated response to the external power and clock control units.

The PCCB does not have a dedicated CHI port. It shares a device port with the HN-D node in the mesh.

If you configure CMN-700 to have multiple asynchronous clock domains, then the PCCB drives one clock signal to each clock domain. For more information about asynchronous clock domain support, see 3.2.1 Clock domain configurations on page 78.

3.1.1.10 System Address Map

All CHI commands must include a fully resolved network address. The address must include a source and target ID. Target IDs are acquired by passing a request address through a *System Address Map* (SAM), which effectively maps a memory or I/O address to the target device.

The SAM functionality is required for each requesting device. The SAM consists of two logical units:

RN SAM

Allows each RN to map addresses to HN-F, HN-I, HN-D, HN-T, HN-P, HN-V, and CCG target IDs.

The RN SAM supports generation of *Memory Controller* (MC) target IDs, which can be used to issue PrefetchTarget operations from the RN directly to the MC.

HN-F SAM and MC SAM

Maps addresses to MC target IDs.

CMN-700 has software-configurable SAM blocks which allow an implementation of CMN-700 to support programmable mappings of addresses to HNs and SNs. See 3.4.4 System Address Map on page 110 for more information about software-configurable SAM blocks.

The SAM functionality is required for each requesting device.

3.1.1.11 Debug and Trace Controller

The *Debug and Trace Controller* (DTC) controls distributed *Debug and Trace Monitors* (DTM) and generates time stamped trace using the ATB interface.

The DTC performs the following functions:

- Generates event or PMU-based interrupts
- Receives packets from DTM and packs them into ATB format trace
- Time stamps trace with SoC timer input
- Generates alignment sync for the ATB trace output
- Handles ATB flush requests
- Handles debug and Secure debug external requests
- Provides a consistent view of distributed and central PMU counters
- Handles PMU snapshot requests
- Generates interrupt INTREQPMU assertion on overflow of PMU counters

3.1.1.12 QoS regulator

CMN-700 supports end-to-end *Quality-of-Service* (QoS) which guarantees using QoS mechanisms that are 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 met
- Use the integrated QoS regulators at ingress points to CMN-700

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 meet this requirement, CMN-700 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. The QR adjusts the QoS field upwards for higher priority in the system and downwards for lower priority.

3.1.1.13 Component Aggregation Layer

A *Component Aggregation Layer* (CAL) allows multiple devices to connect to a single device port on an XP.

CMN-700 has multiple types of CAL. The different types of CAL support connection of different types of devices.

`CAL2, CALBYP2` CAL that can connect to two identical devices. CAL2 supports the following device types:

- RN-F_CHIB_ESAM
- RN-F_CHIC_ESAM
- RN-F_CHID_ESAM
- RN-F_CHIE_ESAM
- RN-I
- RN-D
- HN-F
- HN-I
- HN-P
- SBSX
- SN-F
- CCG

CAL4, CALBYP4 CAL that can connect to four identical devices. CAL4 only supports the following device types:

- RN-F_CHID_ESAM
- RN-F_CHIE_ESAM
- SNF-E

HCAL2 A special CAL that can support 2 different device types. HCAL2 only supports the following device types, but there is no restriction on device ordering when attached to the HCAL2:

- RNI, HNI
- RND, HNI
- RNI, RND

The bypass variants of each CAL type, CALBYP2 and CALBYP4 are the same except for their flit buffering behavior. If there are enough credits for a flit to be sent between the device and MXP, CALBYP2 and CALBYP4 incur no latency when passing flits. However, using CALBYP2 or CALBYP4 increases timing pressure.

CAL2 and CAL4, the variants of each CAL type without bypass functionality, always incur a single cycle of latency, even if there are credits to send flits. This behavior also applies to the MXP to device path.

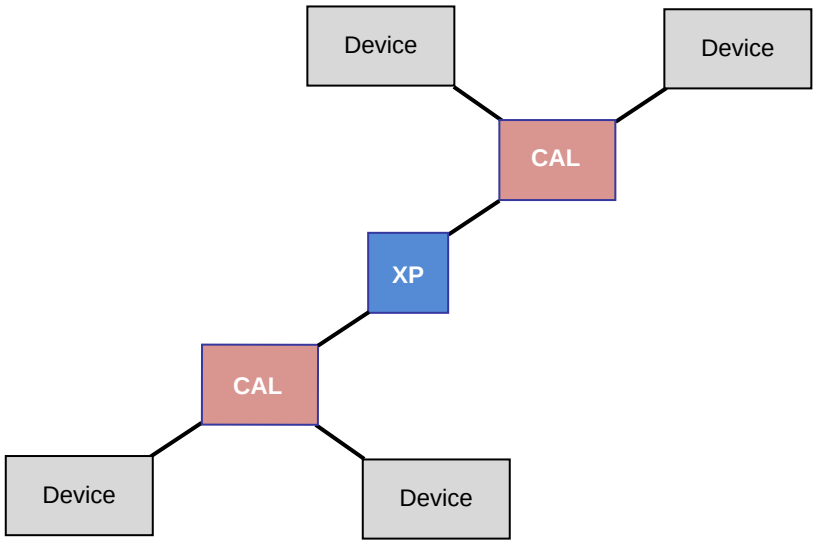
All devices that are connected to a single CAL must be of the same type and must be configured identically.


Note

The HCAL is the exception to this restriction, and the supported device types are previously listed.

The following figure shows a CAL2 example configuration.

Figure 3-5: CAL2 example configuration



3.1.1.14 Credited Slices

You can configure various optional credited register slices in your CMN-700 system. These Credited Slices can help with timing closure.

Credited Slices enable synchronous but higher latency communication at any point in the system.

CMN-700 includes the following optional Credited Slices:

Mesh Credited Slice

Placed between XPs. See 3.1.1.14.1 Mesh Credited Slice on page 52.

Asynchronous Mesh Credited Slice

Placed between XPs that are in different clock domains. For more information, see 3.1.1.14.2 Asynchronous Mesh Credited Slice on page 53.

Device Credited Slice

Placed between a device and a CAL, or a device and an XP. For more information, see 3.1.1.14.3 Device Credited Slice on page 54.

CAL Credited Slice

Placed between a CAL and an XP. See 3.1.1.14.4 CAL Credited Slice on page 54.

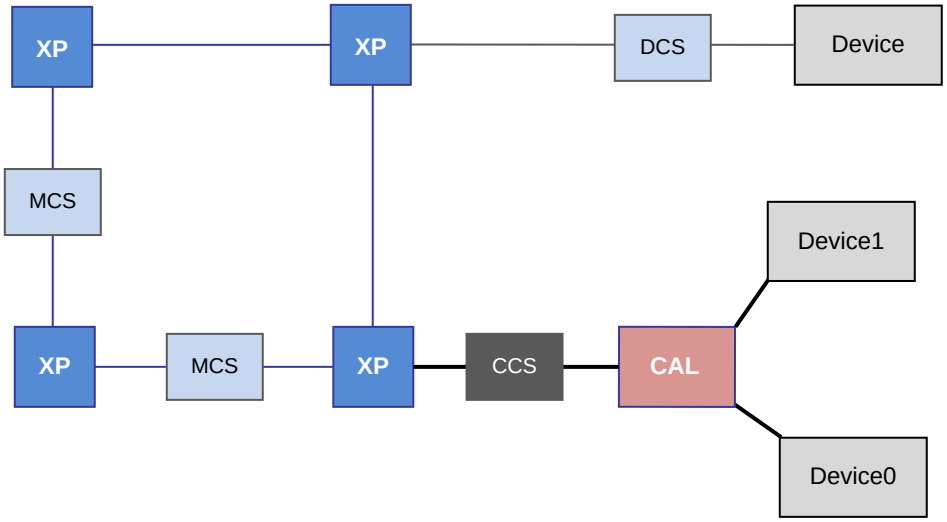
HCAL Credited Slice

Placed between an HCAL and an XP. See 3.1.1.14.4 CAL Credited Slice on page 54.

The slices are simple repeater-flop structures that are applied across the entire communication boundary. The supported number of Credited Slices of each type is specifies the supported number in 2.5.3 Device placement and configuration on page 30.

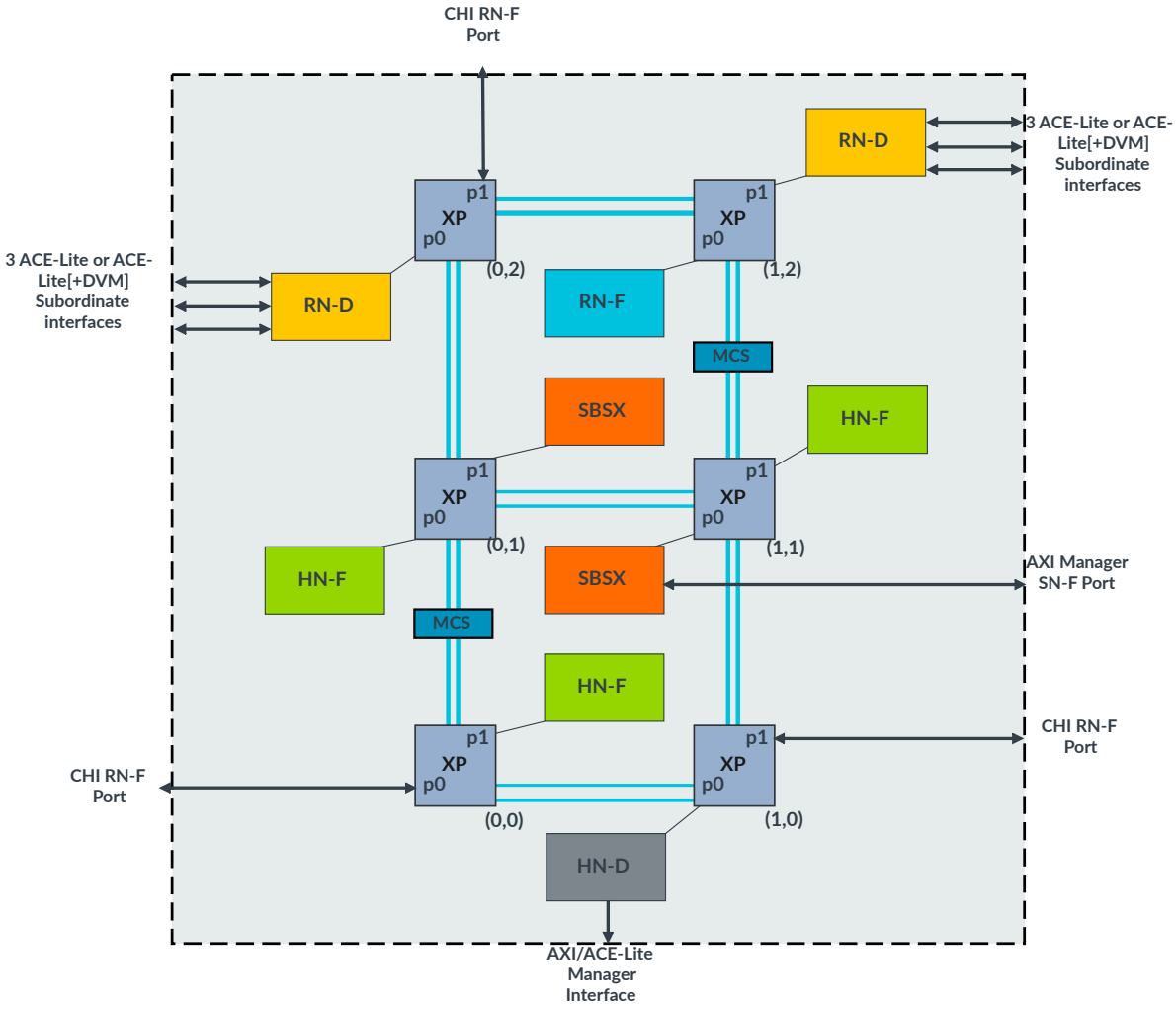
The following figure shows where various Credited Slices fit in the structure of the mesh. The example mesh includes two MCSs, a DCS, and a CCS.

Figure 3-6: Example MSC, CCS, and DCS configuration



The following figure shows a full mesh and device topology with MCS and DCS.

Figure 3-7: CMN-700 system with MCS and DCS



3.1.1.14.1 Mesh Credited Slice

You can configure one or more *Mesh Credited Slices* (MCSs) between CMN-700 XPs. MCSs are optional register slices that can help timing closure in a CMN-700 system.

The CMN-700 mesh can operate with a single cycle of latency between XPs. However, depending on the fabrication process and the distance between XPs, a single-cycle XP-XP connection might limit frequency. In this case, one or more MCSs can be added to lengthen the XP-XP links. Register slices add link transfer latency, but also allow certain CMN-700 implementations to run at higher frequencies.

Each MCS on an XP-XP link adds an extra cycle between XPs. One to four MCSs can be added to any link between XPs.

A configurable buffer depth of 2-4 is allowed and you can select it using the `MCS_RXBUF_NUM_ENTRIES` parameter.

For timing, you can enable an optional register slice on the PUB, internal communications system, interface using the `MCS_PUB_RSL_EN` parameter. By default, register slices are only applied to the CHI interfaces.

3.1.1.14.2 Asynchronous Mesh Credited Slice

CMN-700 supports multiple asynchronous clock domains across the mesh. *Asynchronous Mesh Credited Slices* (AMCSs) perform clock domain crossing between two asynchronous mesh clock domains.

To configure multiple clock domains in the mesh, you must also configure an AMCS on all XP-XP links that span different clock domains. Up to four MCS can be included along with AMCs in any sequence on the XP-XP link.

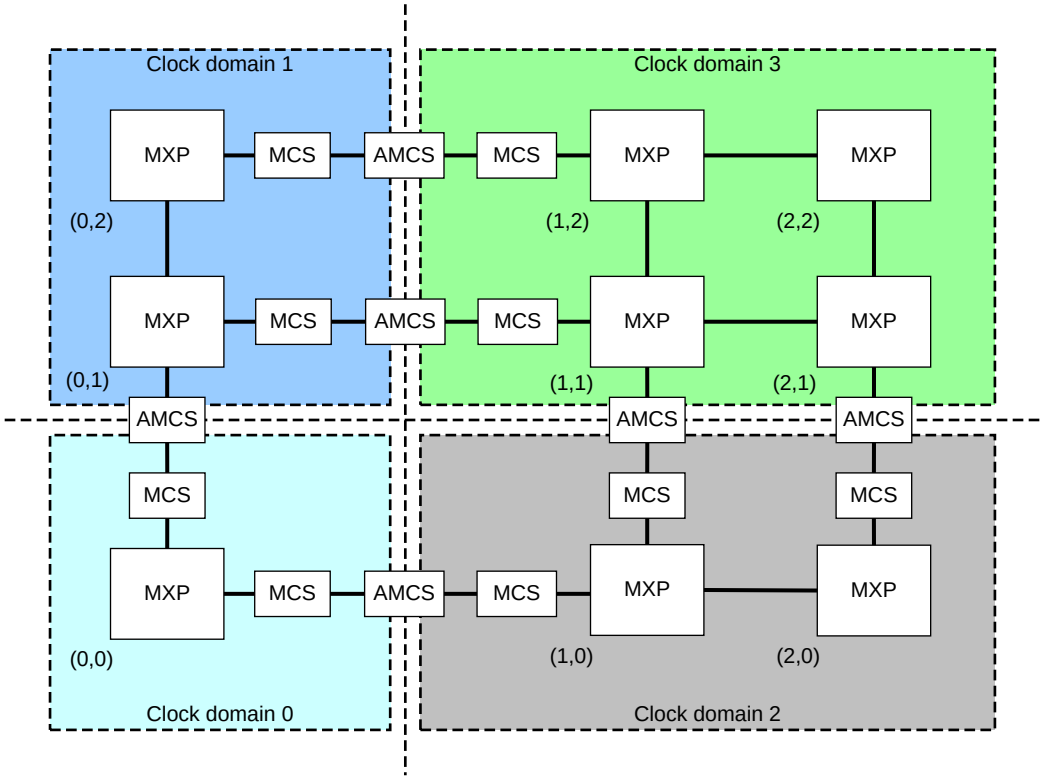
The AMCS does not perform frequency scaling, it only synchronizes traffic between asynchronous clock domains. All CMN-700 clock domains must operate at the same frequency.

When using AMCSs to create multiple asynchronous clock domains, you must divide the mesh into four quadrants, each with a single clock domain. These quadrants must be rectangular and include one or more XPs.

For more information about CMN-700 support for asynchronous clock domains, see 3.2.1 Clock domain configurations on page 78.

The following figure shows an example CMN-700 topology with four asynchronous clock domains that AMCSs bridge. The clock domains must be arranged in the order that the following figure shows. Clock domain 0 must be the bottom-left quadrant.

Figure 3-8: Example asynchronous mesh topology



3.1.1.14.3 Device Credited Slice

You can configure one or more *Device Credited Slices* (DCSs) on a link between a device and an XP. DCSs help with timing closure in a CMN-700 system.

DCSs are optional register slices that you can add to your CMN-700 configuration. You can add up to four DCSs on any link between a device and an XP.

3.1.1.14.4 CAL Credited Slice

You can configure up to four CCSs on the link between a CAL and an XP.

For device ports with an HCAL2 configured, up to four *Hybrid CCSs* (HCCS) can be configured on the link between an HCAL and an XP. The functionality between the CCS and HCCS is the same.

3.1.1.15 AMBA Domain Bridge

The *AMBA Domain Bridge* (ADB) bridges two AXI, ACE5-Lite, or ACE5-Lite-with-DVM interfaces that operate in two different clock domains, power or voltage domains, or both.

For more information about the ADB, see the *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.

3.1.1.16 CHI Domain Bridge

The *CHI Domain Bridge* (CDB) bridges two CHI interfaces that operate in two different clock domains, power or voltage domains, or both.

For more information about the CDB, see the *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.

3.1.1.17 CXS Domain Bridge

The CXS Domain Bridge (CXSDB) bridges two CXS interfaces that operate in two different clock domains, power/voltage domains, or both.

For more information about the CXSDB, see the *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.

3.1.1.18 Backward compatible RN-F support

CMN-700 is compliant with CHI-E, but can also contain RN-Fs that comply with CHI-B, CHI-C, and CHI-D. Certain restrictions apply to how CMN-700 handles transactions that are sent from older RN-Fs to maintain backwards compatibility.

The following table shows how all CMN-700 blocks handle specific backward compatible CHI-B, CHI-C, CHI-D, and CHI-E features.

Table 3-1: All blocks backward compatibility

All blocks	CHI-B	CHI-C	CHI-D	CHI-E
DBID[9:8]	Must be `0b00`	Must be `0b00`	Can be used	Can be used
DBID[11:10]	Must be `0b00`	Must be `0b00`	Must be `0b00`	Can be used
SNP and DMT REQ TxnID[9:8]	Must be `0b00`	Must be `0b00`	Can be used	Can be used
SNP and DMT REQ TxnID[11:10]	Must be `0b00`	Must be `0b00`	Must be `0b00`	Can be used

If RN-F TxnID[9:8] = `0b11`, DCT from CHI-B, CHI-C, or CHI-D RN-F to CHI-E RN-F cannot be done.

The following table shows how CMN-700 HN-Fs handle specific backward compatible CHI-B, CHI-C, CHI-D, and CHI-E features.

Table 3-2: HN-F backward compatibility

HN-F protocol	CHI-B	CHI-C	CHI-D	CHI-E
Requests from RN-F	Supported	Supported	Supported	Supported
DMT	Yes	Yes	Yes	Yes
DCT	Yes	Yes	Yes	Yes
Separate Response and Data	No	Yes	Yes	Yes
SnpPreferUnique	No	No	No	Yes
SnpQuery	No	No	No	Yes
New fields	MXP drives fixed values	MXP drives fixed values	MXP propagates new fields drives fixed values	MXP to propagate

The following table shows how CMN-700 DNs handle specific backward compatible CHI-B, CHI-C, CHI-D, and CHI-E features.

Table 3-3: DN backward compatibility

DN protocol	CHI-B	CHI-C	CHI-D	CHI-E
Requests from RN-F	Supported	Supported	Supported	Supported
CompDBID for DVM operations	No	No	Yes	Yes

In a system with heterogeneous components, the system configuration must determine the lowest common DVM specification that is supported in the system. In such a system, it might be necessary to set the enable_8_4_termination bit in the por_dn_cfg_ctl register to 1.

If enable_8_4_termination is set to 1, the DN detects Arm®v8.4-A DVM operations, it suppresses their propagation, and responds in a protocol-compliant manner. This behavior avoids deadlocks and denial of service.

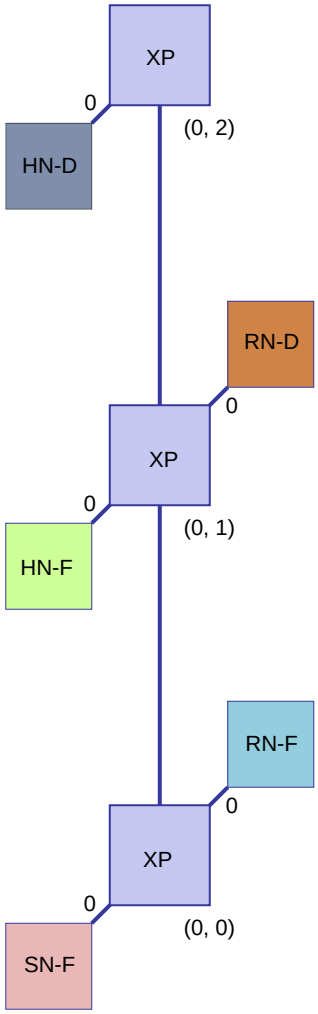
The DN does not send any error indication or log errors when Arm®v8.4-A DVM operations are terminated.

3.1.2 System configurations

CMN-700 can be configured to meet system requirements.

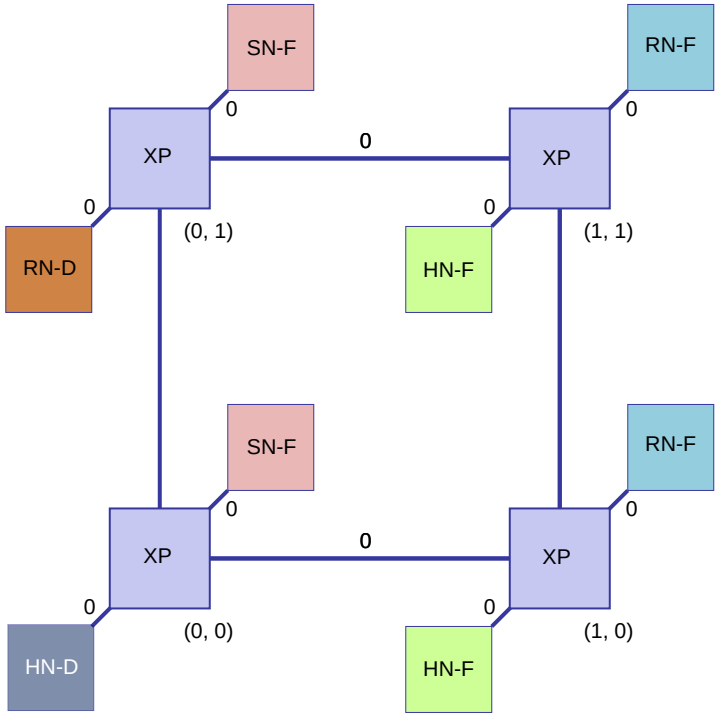
The following figure shows a 1 × 3 mesh for a small system configuration that contains single instances of RN-F, HN-F, RN-D, SN-F, and HN-D.

Figure 3-9: Mesh example 1 × 3



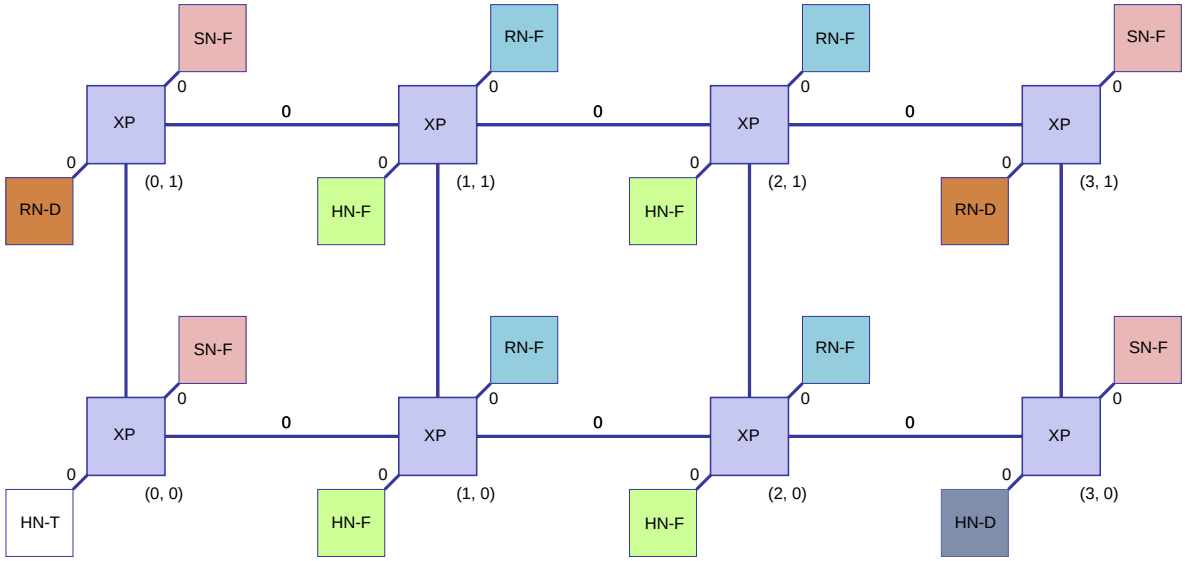
The following figure shows a 2 × 2 mesh for a medium system configuration with single and multiple instances of RN-F, HN-F, RN-D, SN-F, and HN-D.

Figure 3-10: Mesh example 2 × 2



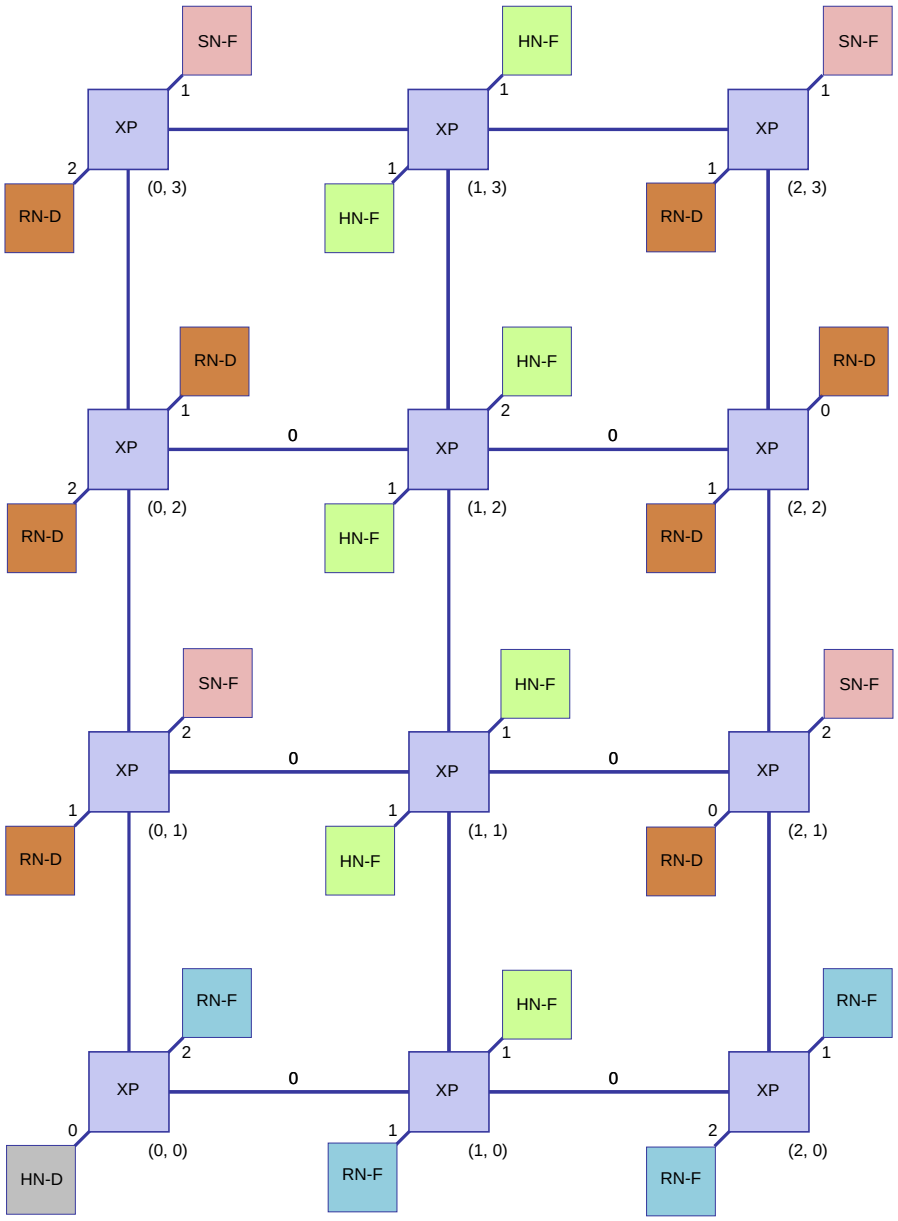
The following figure shows a 4 × 2 mesh for a medium system configuration with single and multiple instances of RN-F, HN-F, RN-D, SN-F, and HN-D.

Figure 3-11: Mesh example 4 × 2



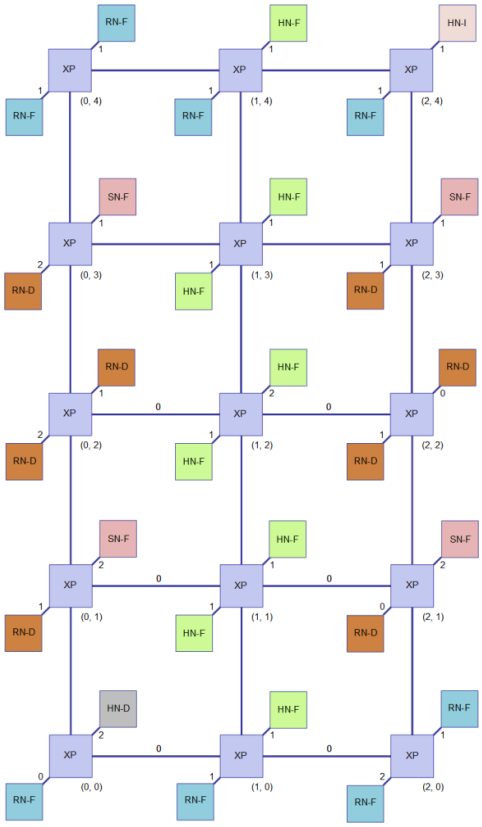
The following figure shows a 3 × 4 mesh for a large system configuration with multiple instances of RN-F, HN-F, RN-D, SN-F, and HN-D.

Figure 3-12: Mesh example 3 × 4



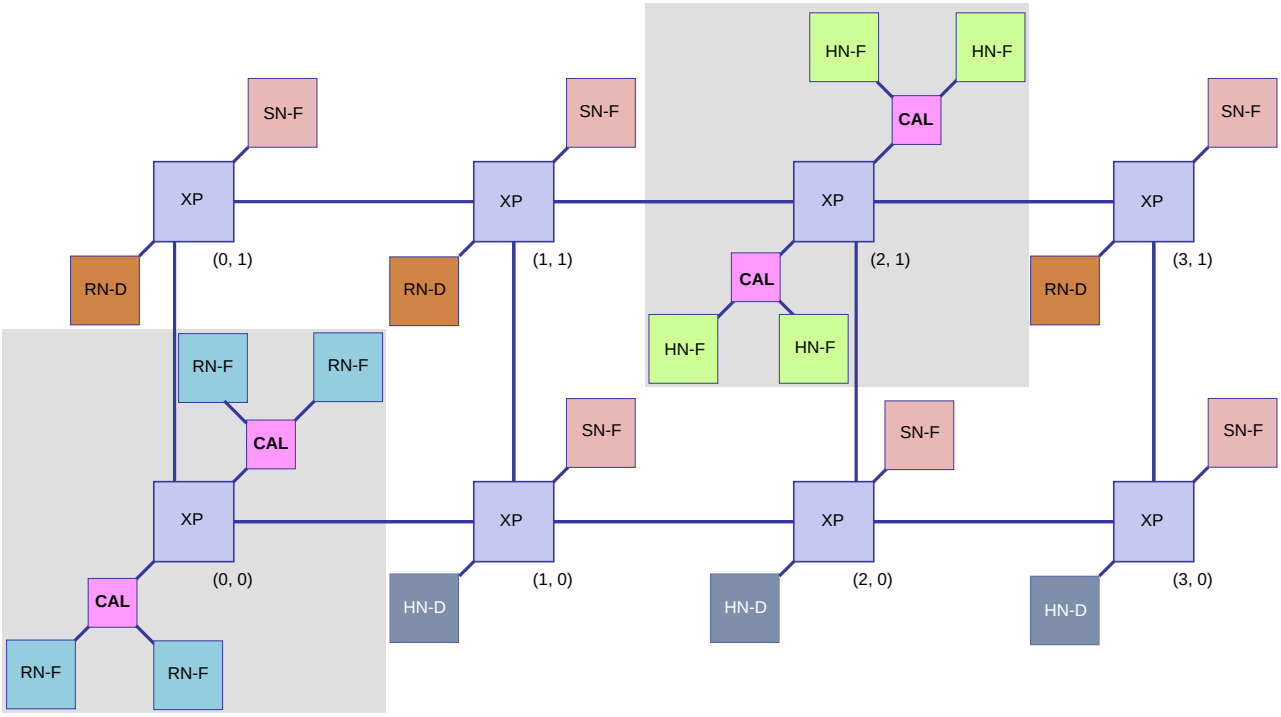
The following figure shows a 3 × 5 mesh for a large system configuration with multiple instances of RN-F, HN-F, RN-D, HN-I, SN-F, and HN-D.

Figure 3-13: Mesh example 3 × 5



The following figure shows a 4 × 2 mesh for a medium system configuration with RN-F and HN-F CAL, that the gray areas shows.

Figure 3-14: Mesh example 4 × 2 with CAL

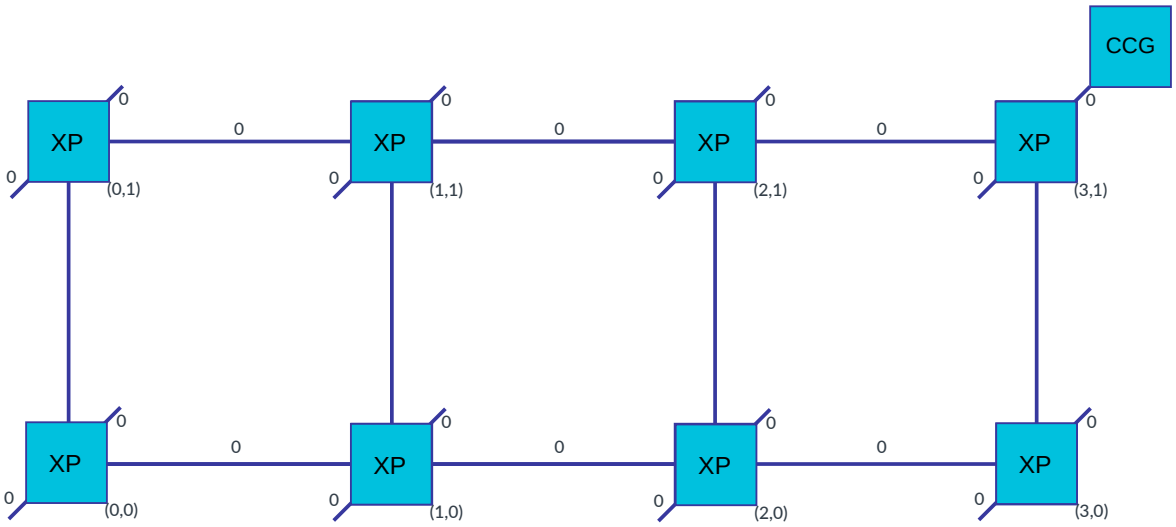


3.1.3 CML system configurations

The following provides examples of CML system configuration that can be used in CMN-700.

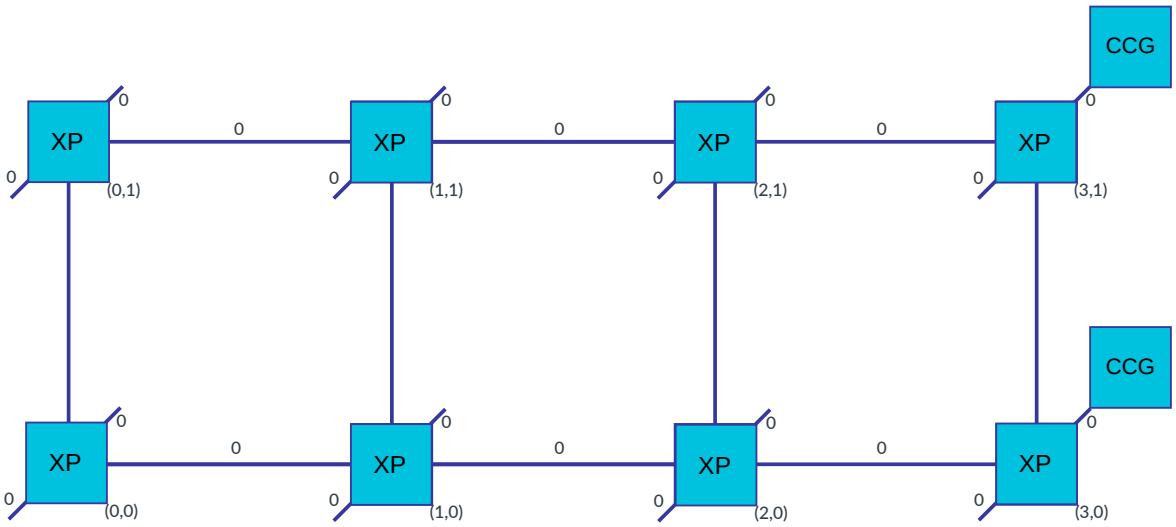
The following figure shows a 4 × 2 single CML mesh example.

Figure 3-15: 4 × 2 single CML mesh example



The following figure shows a 4 × 2 double CML mesh example.

Figure 3-16: 4 × 2 double CML mesh example



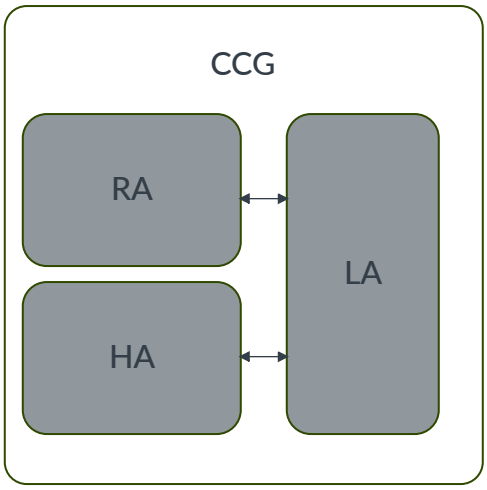
Warning

A CCG node cannot be placed on XP (0,0).

CML components

The following figure shows a CCG block diagram with RA, HA, and LA.

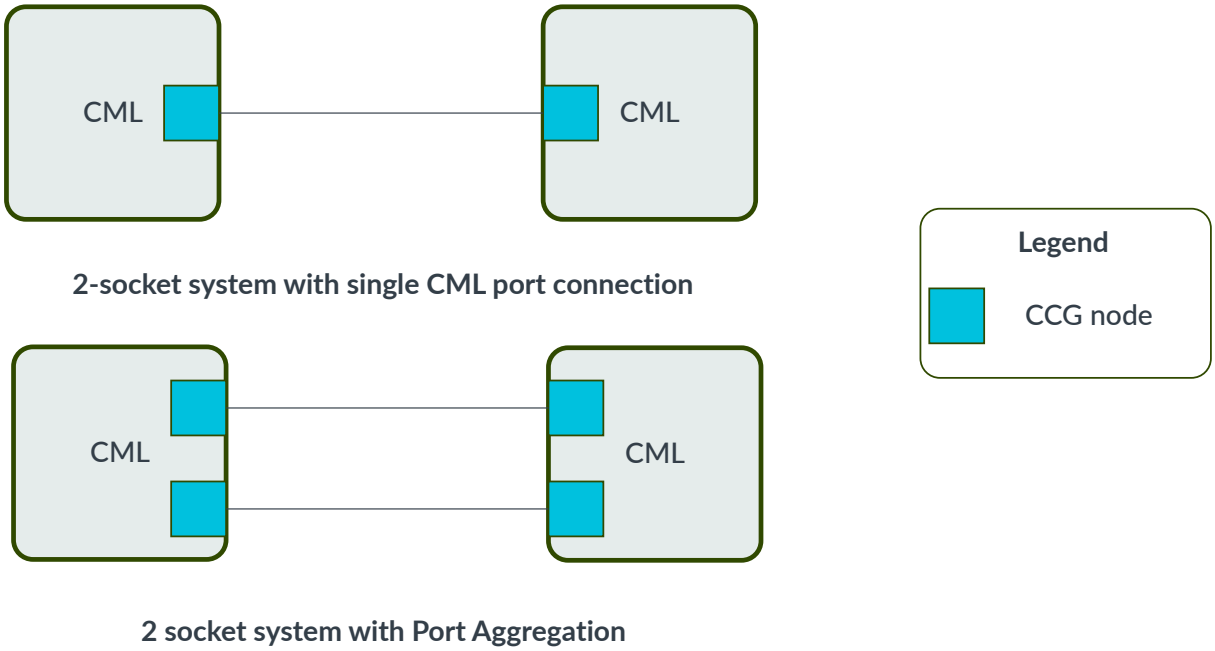
Figure 3-17: CCG block diagram with RA, HA, and LA



CML topologies

The following figure shows three simplified CML topology examples.

Figure 3-18: CML topologies



3.1.3.1 CML Symmetric Multiprocessor (SMP) support

A *Symmetric Multiprocessor* (SMP) allows for a shared, common OS and memory to operate on multiple chips.

CMN-700 supports SMP systems when the systems are built using the same version of CMN-700. For SMP systems, CCG block is required to enable multi-chip SMP communications over a CXS issue B interface.

When set, the provided SMP mode option enables DVM, GIC-D, Exclusives, MPAM, CPU-Event, and selective CHI-D and CHI-E features across CML_SMP links using a micro-architected mechanism.

Trace Tag, when set on CHI request or snoop, is preserved across CML_SMP link.

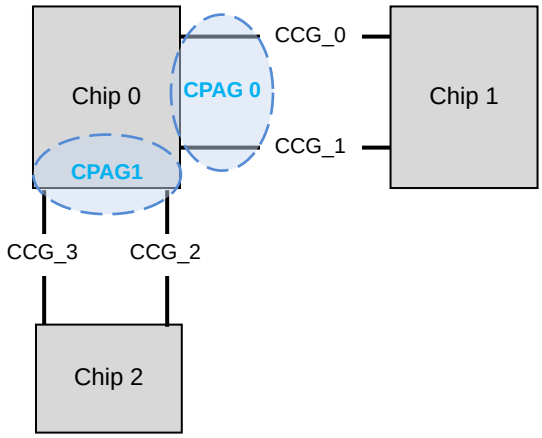
For more information about programming CMN-700 for SMP mode, see 4.4.7 CML programming on page 1134.

3.1.3.2 CML Port Aggregation Groups

The CMN-700 CML configuration supports up to 32 gateway blocks (CCGs). These CML gateways can be grouped in up to 16 *CML Port Aggregation Groups* (CPAGs).

This feature can be used when connecting two or more chips together with multiple ports between the chips. For example, the following figure shows three chips that are connected by four CCGs that are grouped into two CPAGs.

Figure 3-19: CML Port Aggregation Groups



In the preceding example, the following CPAGs are present:

CPAG_0 CPAG with two CML ports (CCG_0 and CCG_1) to connect Chip 1 to Chip 0
CPAG_1 CPAG with two CML ports (CCG_2 and CCG_3) to connect Chip 2 to Chip 0

To enable CPAG, both RN SAM and HN-F registers must be programmed accordingly in each chip.

3.1.3.3 CML credit requirements

Each CML port requires a minimum of one request and one data credit more than the total number of reservations. This number is enabled through the configuration control register of the RA (por_ccg_ra_cfg_ctl).

This requirement applies to each enabled CML_SMP link end at a given CML port. For example, by default all the reservations are enabled in the SMP mode. Therefore, a minimum of four request and four data credits must be granted per CML link end. These credits are used by certain traffic types, such as QoS-15 or PCIe, to progress in a loaded system. See 4.3.3.5 por_ccg_ra_cfg_ctl on page 332.

3.1.3.4 Tunneling PCIe traffic through CML SMP link

CMN-700 supports tunneling of PCIe traffic targeting remote memory.

Tunneling of PCIe traffic is advantageous if the local and remote memory is high ordered interleaved (for example NUMA). An address range is considered to be high order interleaved if the address range programmed in a RN SAMs hashed or non-hashed address region is either targeting remote memory or local memory but not both. An address range is considered low ordered interleaved if the programmed address range is striped across both local and remote HNs. Usually low order interleaving using RN SAMs same address region is only expected for HN-F bound memory.

To enable this function, the CCG on the remote chip must contain an RN-I. You achieve this by setting the CCG `PCIE_ENABLE` configuration parameter to true. You must program local PCIe RN-I, RN-D, and CCG blocks accordingly to enable this traffic. A PCIe RN-I or RN-D has PCIe IP connected to an ACE-Lite interface, and has the pcie_mstr_present configuration register bit set.

3.1.3.4.1 PCIe writes

To optimize PCIe write bandwidth targeting remote memory, a PCIe RN-I or RN-D sends these writes to CCG in pipelined fashion.

The CCG also issues early completion for intermediate (non-last) data beats for write burst requests originating from the same AXI burst sent to the PCIe RN-I or RN-D. This behavior helps to reduce tracker lifetime and increases the number of outstanding requests from the PCIe RN-I or RN-D.

If these writes are targeting a remote PCIe EP through HN-P, then CCG also preserves additional information across SMP link. This behavior helps with gathering of these writes downstream of remote HN-P. To enable write gathering at the endpoint, the CCG creates uniquely identifiable write streams coming from different RN-Is and RN-Ds in the system. To create unique streams, the CCG sends the following information to the remote CCG RN-I:

- The Chip ID

- Logical RN-I/D (RAID)
- RN-I/D AXI port ID
- And AXI AWID information received from the issuing PCIe RN-I or RN-D

Remote CCG RN-I then issues a request to HN-P with all the identifying information that can be used downstream of HN-P to gather write bursts.

For more information about RN-I and RN-D write burst cracking and write gathering flow, see 7.1.1 RN-I and RN-D write Burst cracking on page 1228.

If an RN-I is not present at remote CCG, then write tunneling must be disabled. By default, write tunneling is enabled.

Write tunneling in CCG can be disabled by setting the dis_rnid_tnl_retry_trk and dis_rnid_early_wrcomp bits, and clearing the remote_rni_present bit in the por_ccg_ra_aux_ctl register.

Write tunneling at PCIe RN-I and RN-D can be disabled by setting the dis_pci_cxra_logical_tgt_2hop bit in the por_rni_aux_ctl register.

To uniquely identify the originating PCIe RN-I and RN-D for write gathering, a unique RAID must be assigned to each PCIe RN-I and RN-D. These RAIDs are required to be unique only accross PCIe RN-I and RN-D. There is also a software controlled CHIPID that is used to uniquely identify the PCIe RN-I and RN-D.

If the target is remote HN-P, *CML Port Aggregation* (CPAG) must be disabled to preserve write ordering

3.1.3.4.2 PCIe reads

Read bursts coming from PCIe RN-I and RN-D targeting PCI_CXRA node types are preserved across CML SMP link.

If the remote target is an HN-P, the remote CCG RN-I then further issues a read burst to HN-P. If the remote target is an HN-F, the remote CCG RN-I cracks the read bursts into 64B chunks.


Note

If an RN-I is not present on the remote side (that is at remote CCG), PCIe Read bursts cannot be issued to a CCG.

To indicate that a remote CCG RN-I is present, set the remote_rni_present field of the por_ccg_ra_aux_ctl register.

For more information abour RN-I and RN-D read burst preservation, see 7.1.3 PCIe read Burst preservation through the interconnect on page 1229.

For more information about RN-I and RN-D support for AXID-based targetID selection for HN-P targets for both PCIe reads and writes, see 7.1.4 RN-I and RN-D AxID-based target selection on page 1230.

For further programming requirements for tunneling PCIe traffic, see 4.4.7 CML programming on page 1134.

3.1.3.5 Streaming PCIe write traffic through CML SMP link

CMN-700 supports streaming of PCIe ordered writes when targeting remote memory. This makes use of Optimized Streaming of Ordered Write mechanism as defined in AMBA 5 CHI specification.

This mechanism is used when the address range targeted by PCIe traffic, programmed in a given RN SAM address region, is striped across local and remote HNs at certain interleaving granularity. In CMN-700, the interleaving granularity, below which RN-I/D will use this streaming mechanism, is set to 4K. Additionally, this mechanism can be used if PCIe write traffic cannot be tunneled through a given CCG. For information about when tunneling is disabled, see 3.1.3.4 Tunneling PCIe traffic through CML SMP link on page 66.

For programming requirements to enable PCIe write traffic, see 4.4.7.3.14 Programming to enable PCIe write streaming through CML SMP link on page 1146.

3.1.4 Structural configuration and considerations

CMN-700 supports various options that you can use to further configure the structure of the interconnect. Some of these options have specific restrictions and requirements, and modify the behavior of the interconnect from the default behavior.

CMN-700 supports the following structural options:

- Dual DAT, REQ, RSP, and SNP channels
- Dedicated RN-I resource for traffic through AXI ports
- *CML Port Aggregation Groups* (CPAGs)
- Using extra device ports on MXPs
- Connection of devices with multiple interfaces

3.1.4.1 Dual CHI channels

To reduce congestion within the mesh, CMN-700 provides a configuration option to double the number of CHI channels. This feature is useful if you have heavy traffic sources that saturate specific routing paths.

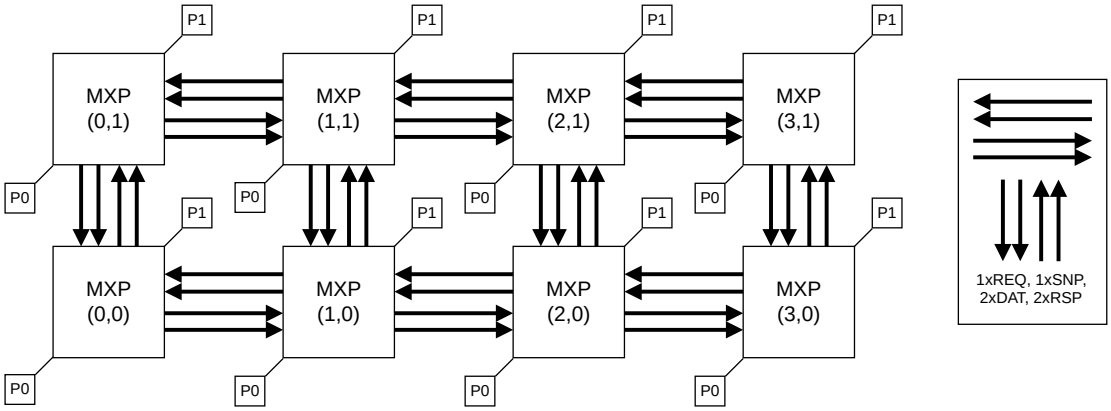
If the `EN_2X_[DAT/RSP]_VC` parameter is enabled, the number of DAT and RSP CHI channels per MXP-MXP connection increases from one to two. Each channel can be controlled independently. This option has the following benefits:

- Increases device upload bandwidth
- Increases MXP-MXP bandwidth

The `EN_2X_REQ_VC` and `EN_2X_SNP_VC` parameters, when enabled, add an additional pair of REQ and SNP CHI channels.

The following figure represents the dual CHI channels in the mesh.

Figure 3-20: Dual DAT and RSP channel topology



Boot-programmable registers determine which of the dual channels to select at flit upload. For more information about configuring the selection, see 3.1.4.3 Dual CHI channel selection registers on page 71.

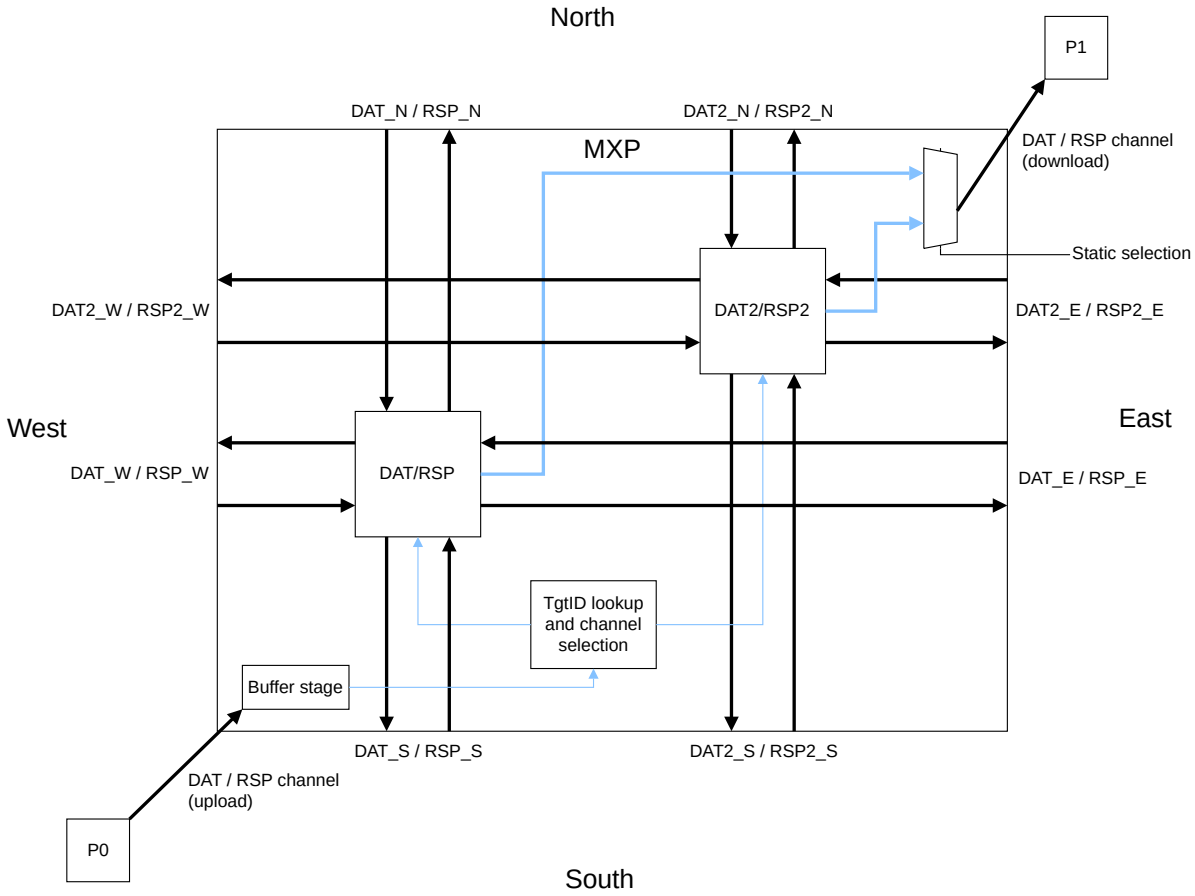
For more information about the procedure to program these registers, see 4.4.4 Program the dual DAT and RSP channel selection scheme on page 1132.

3.1.4.2 Dual CHI channel selection

If you enable the dual CHI channel feature, CMN-700 uses a TargetID-based selection scheme to determine which channel to upload CHI flits to for the DAT, RSP, and SNP CHI channels. The REQ channel uses a SourceID-based selection scheme.

The following figure shows the channel selection mechanism in the MXP. The figure shows the upload channel selection mechanism at port 0 and the download channel selection mechanism at port 1. However, port 0 can use the download channel selection mechanism and port1, can use the upload channel selection mechanism. The mechanisms are similar for the DAT, RSP, and SNP CHI channels.

Figure 3-21: Dual CHI channel selection mechanism



Each mesh target is associated with one of the two DAT channels. All traffic to a single target is mapped to one of the two DAT channels. Likewise, all responses and snoops are mapped to one of the two RSP or SNP channels respectively.

The REQ channel differs from the other CHI channels because it uses the SourceID to route to one of the two REQ channels. This is done because RNF device types that send requests into the MXP can have their TargetID overriden when an address lookup is performed. This change in channel selection behavior also allows for traffic to a single target to be received from both REQ channels.

When you enable this feature by setting the configuration parameter, CMN-700 applies a default channel selection scheme. Otherwise, traffic to targets on MXPs with an even XID uses channel 0 and traffic to targets on MXPs with an odd XID uses channel 1. This scheme is active until you specify a new scheme at boot by programming the channel selection behavior. See 4.4.4 Program the dual DAT and RSP channel selection scheme on page 1132.

The `MESH_2X_DEF_SEL` parameter value affects the default selection scheme in the following way:

0 Default mesh channel selection scheme depends on whether the tgt XID is even or odd. For 1 x N mesh configurations (one column), traffic to all targets in the network use channel 0.

1 Default mesh channel selection scheme depends on whether the tgt YID is even or odd. For N x 1 mesh configurations (one row), traffic to all targets in the network use channel 0.

After you have programmed the scheme, the MXP compares the TargetID of uploaded flits to the programmed *Lookup Table* (LUT). This LUT specifies which channel the MXP must send the flit to.

3.1.4.3 Dual CHI channel selection registers

Boot-programmable registers define the TargetID-based LUT. The MXP uses this LUT to select CHI channels for flits on the DAT, RSP, and SNP channels. The REQ channel only uses SourceID XID or YID based routing.

If the dual CHI channel feature is enabled, using por_mxp_multi_mesh_chn_sel_*, then each MXP contains 16 64-bit registers. The contents of these registers define the LUT that selects which channel a flit is assigned to. Each register can specify the assignment channel for four TargetIDs, so you can configure the behavior for up to 64 targets overall.

After the dual CHI registers have been programmed, any remaining unprogrammed TargetIDs are automatically assigned to channel 0. Unprogrammed registers and fields hold the reset value and are not part of the lookup. Programmed registers and fields are only valid for lookup when the multi_mesh_chn_sel_reg_*_valid bit of the register is set.

The following table shows the register format. Each 11-bit TargetID + CHN_SEL field specifies a TargetID and the CHI channel for flits with that TargetID.

Table 3-4: por_mxp_multi_mesh_chn_sel_N (N=0-15) register format

Bit field	Description
[63]	VALID
[62:48]	2-bit CAL_DEV_CHN_MAP_SEL + 11-bit TargetID + CHN_SEL
[47:32]	
[31:16]	
[15:0]	

Each 11-bit TargetID + CHN_SEL field has a reset value of `11'b0`. One or more fields can be left to hold the reset value in a programmed register with the multi_mesh_chn_sel_reg_*_valid bit set. In this case:

- Fields containing the reset value map TargetID 0 to channel 0 by default.
- If TargetID 0 is programmed to map to channel 1 by one of the register fields, then that value determines the mapping. This mapping is used even if there are unprogrammed fields in the register.

The following table shows the channel map select, for target devices behind CAL that are associated with the corresponding TargetID fields.

Table 3-5: CAL_DEV_CHN_MAP_SEL corresponding TargetID fields

TargetID	CAL	Description
`2'b00`	CAL2	All devices behind CAL are mapped to the same channel in the multi-channel Mesh structure as the following specifies, applicable for Mesh with > 2 device ports per XP: • CAL2: DEV0, DEV1 is mapped to the same channel
`2'b01`	CAL4	All devices behind CAL are mapped to same channel in the multi-channel Mesh structure as the following specifies: • DEV0, DEV1, DEV2, DEV3 are mapped to the same channel
`2'b10`	Reserved	-
`2'b11`	CAL2, CAL4	Not supported in CMN-700

The MXP also contains a control register, por_mxp_multi_mesh_chn_ctrl. Program this register to indicate that the dual CHI channel selection configuration is complete.

The following table shows the control register format.

Table 3-6: por_mxp_multi_mesh_chn_ctrl register format

Bit field	Description
[63:1]	Reserved
[0]	multi_mesh_chn_sel_programmed

For more information about programming these registers, see 4.4.4 Program the dual DAT and RSP channel selection scheme on page 1132.

Replicated channels

You can enable the 2xMesh in you configuration by using the following global user parameters:

- `POR_2XREQ_EN_PARAM`
- `POR_2XSNP_EN_PARAM`
- `POR_2XRSP_EN_PARAM`
- `POR_2XDAT_EN_PARAM`

If RNFE or RCCAL are in the configuration, then the configuration is considered a 2xMesh.

Modes supported by configuring each device port using por_mxp_device_prt_ctl.p[0/1/2/3/4/5]_multi_chan_sel mode registers:

- 2'h0: Enable channel mapping based on TGTID scheme
- 2'h1: Enable channel mapping based on based on dynamic credit availability scheme
- 2'h2: Enable channel mapping based on direct connect scheme

Modes supported per device type

RNFE

Direct Connect Scheme only

RCCAL

TGTID Based Selection Scheme

Dynamic Credit Availability based selection scheme

SNFE

TGTID Based Selection Scheme

Dynamic Credit Availability based selection scheme

All other devices [RN*/CCRA/HN*/CCHA]

TGTID Based Selection Scheme only

3.1.4.4 Dedicated RN-I resources for AXI port traffic

You can set up dedicated resources per AXI port in the RN-I and RN-D. Using this feature, you can ensure that one manager that is connected to and RN-I or RN-D does not block progress of traffic from another manager.

For example, you can use this feature to connect SMMU and GIC components to RN-I or RN-D AXI ports, and ensure that real-time traffic from both components can progress.

To support this feature, the RN-I and RN-D have two sets of configuration registers per port and an auxiliary control register. You must program these registers to enable this feature.

Programming the registers for a port reserves a specific number of tokens for that port. When these tokens are reserved, the manager that is attached to that port is guaranteed an equivalent number of tracker entries and corresponding resources. Other managers that are connected to the RN-I or RN-D are blocked from using these reserved resources. Therefore traffic from the manager with reserved tokens can progress. You can divide up RN-I or RN-D resources between different ports according to the relative resource requirements of different managers.

The following table shows the configuration registers that you program to enable this feature. The table also shows the associated register fields and some properties of those fields.

Table 3-7: Configuration register fields for dedicated GIC and MMU RN-I resources

Register name	Register field	Bits	Reset value	Description
por_rni_cfg_ctl	dis_port_token	[6]	`1'b1`	Enables and disables per port reservation for all ports for both read and write channels. Disables QoS15 reservation. **Note:** CR_QPC_EN_Q enables QoS15 reservation.
por_rni_s0_port_control	s0_rd_token	[17:11]	`7'h0`	Number of reserved read tokens for port 0, per slice
por_rni_s0_port_control	s0_wr_token	[24:18]	`7'h0`	Number of reserved write tokens for port 0, per slice
por_rni_s1_port_control	s1_rd_token	[17:11]	`7'h0`	Number of reserved read tokens for port 1, per slice
por_rni_s1_port_control	s1_wr_token	[24:18]	`7'h0`	Number of reserved write tokens for port 1, per slice
por_rni_s2_port_control	s2_rd_token	[17:11]	`7'h0`	Number of reserved read tokens for port 2, per slice

Register name	Register field	Bits	Reset value	Description
por_rni_s2_port_control	s2_wr_token	[24:18]	7'h0	Number of reserved write tokens for port 2, per slice

For each port, software can program the number of tokens that correspond to the number of reserved entries in the tracker per slice. The token fields are 7 bits wide to accommodate the maximum tracker size of 96 entries. For a smaller tracker size, only the appropriate bit ranges are used. For example, for a 16-entry tracker, only bits [3:0] of the 7-bit field are used to indicate the number of reserved tokens.

When you enable this feature using the dis_port_token field of the por_rni_cfg_ctl register, QoS15-based reservation is disabled.

The number of reserved entries per port is equal to the programmed token value plus one. For example:

If s0_rd_token = 3

Four tracker entries are reserved

If s0_rd_token = 0

One tracker entry is reserved

If the total number of the programmed reserved entries from all ports is larger than the number of read tracker entries in the slice minus 1, the number of tokens has been misprogrammed. To prevent overflow because of misprogramming, the RN-I reserves a specific number of tokens for each port. The RN-I uses a quarter of the programmed value of each port to calculate the number of reserved entries. If this value is a fraction, then the remainder is rounded to 0.

Example 3-1: Programmed values for eight-entry tracker

Consider the following programmed values for a tracker with eight entries:

s0_port_token = 4

Reserving five entries

s1_port_token = 0

Reserving one entry

s2_port_token = 1

Reserving two entries

In this case, the total number of reserved entries is eight, and the tracker has no unreserved entries. However, the number of tracker entries minus one is seven, and so the registers are misprogrammed. In this case, the RN-I divides each register value by four to calculate the final number of reserved entries:

s0 = 4/4 = 1

Two entries are reserved

s1 = 0/4 = 0

One entry is reserved

s2 = 1/2 = 0

One entry is reserved

Therefore, a total of four entries are reserved for this configuration. The tracker has four unreserved entries.

3.1.4.5 Support for extra device ports on MXPs

You can expand the number of device ports on an MXP beyond the default number of two. This feature reduces the power consumption of CMN-700 configurations by reducing mesh transfer activity between MXPs.

The maximum permitted number of device ports per MXP is configuration-dependent.

Enabling extra device ports affects the following aspects of CMN-700:

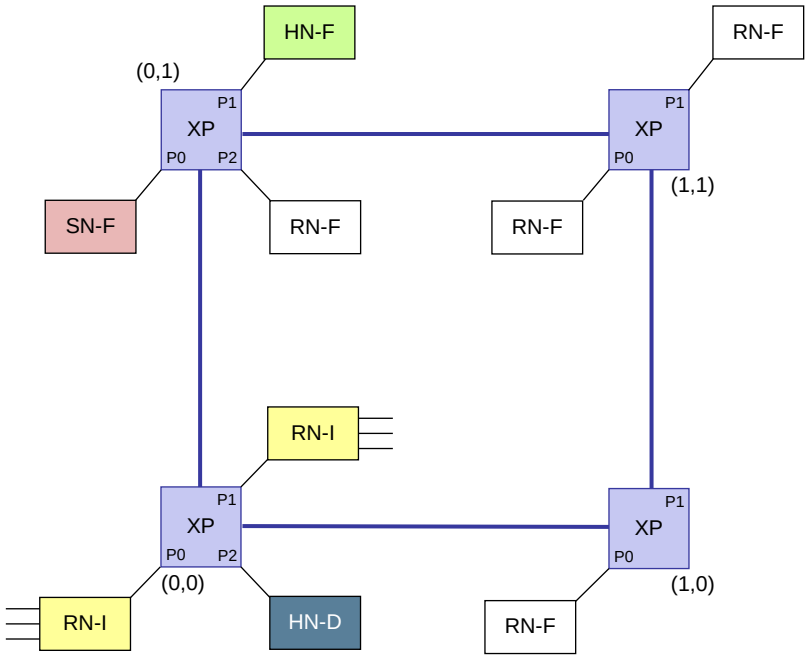
- The overall system topology. See 3.1.4.6 Topology considerations when using extra device ports on page 75.
- The number and mapping of DTMs in the MXP. See 3.1.4.7 DTM changes when using extra device ports on page 76.
- The mapping scheme for node IDs. See 3.4.2 Node ID mapping for configurations with extra device ports on page 106.

3.1.4.6 Topology considerations when using extra device ports

The number of devices in your CMN-700 design determines the number of MXPs and device ports you require. The design is also affected by whether you choose to put a CAL on any of the device ports.

The following figure shows an example mesh configuration using extra device ports.

Figure 3-22: Mesh configuration using extra device ports



3.1.4.7 DTM changes when using extra device ports

A configuration parameter determines whether DTMs are replicated in MXPs to support extra device ports. Each port is mapped to a specific DTM and both the configuration parameter value and the overall system configuration determines how the ports are mapped.

Note

For a mesh configuration, up to four device ports are permitted per MXP. For more information, see 3.1.4.6 Topology considerations when using extra device ports on page 75.

To enable support for multiple DTMs in MXPs, set the `MXP_MULTIPLE_DTM_EN` parameter = 1. If this parameter is set to 1, then the DTM in each MXP is replicated according to the number of device ports on the MXP. Each DTM supports up to two device ports. The different interconnect configurations use the following DTM mappings:

Mesh configuration

- DTM0 supports P0 and P1
- DTM1 supports P2 and P3

If the `MXP_MULTIPLE_DTM_EN` parameter = 0, each MXP has a single DTM. In this case, for mesh configurations the single DTM supports P0, P1, P2, and P3.


Note

If a single DTM supports more than two ports, it might not be possible to monitor all the events on different ports at the same time. However, using a single DTM saves on the implementation area.

3.1.5 DSU and DMC AXI5 Utility Bus

CMN-700 provides a mechanism to interface with an *AXI5 Utility Bus* (AXU), which is a simplified version of AXI5.

The accesses can be either 32b or 64b in size, and single data transfer with no burst. The AXU is connected to RN-F and SN-F node locations as an AXI5 manager port.


Note

AXU is configurable for RNF-*ESAM and SNF-E locations only

Each AXU port has a logical ID. One set of logical IDs are contiguously numbered for RN-F, and another set are contiguously numbered for SN-F. To send a request to those ports, software must look up the logical ID base information from por_mxp_p<0..5>_info.dsu_logicalid_base_p<0..5> and por_mxp_p<0..5>_info.dmc_logicalid_base_p<0..5>.

The DSU address region starts from `DSU_PERIPHBASE`. Each DSU occupies a fixed 1MB space. Up to 256 DSUs can be supported. You can calculate the total DSU address space, in MB, using the total number of DSUs rounded up to next power of 2. A base aligned with this address space must be driven to `DSU_PERIPHBASE`. The DSU address is composed of [LSB of `DSU_PERIPHBASE` -1:20][19:0]. The lower 20 bits is the DSU offset, and the higher bits form the logical ID.

The DMC address region starts from `DMC_PERIPHBASE`. Each DMC occupies a fixed 16MB space. Up to 16 DMCs can be supported. You can calculate the total DMC address space, in MB, using the total number of DMCs multiplied by 16. A base aligned with this address space must be driven to `DMC_PERIPHBASE`. The DMC address is composed of [LSB of `DMC_PERIPHBASE`-1:24][23:0]. The lower 24 bits is the DMC offset, and the higher bits form the logical ID.

To facilitate APB access, we recommend that all three bases,`CFGM_PERIPHBASE`, `DSU_PERIPHBASE`, and `DMC_PERIPHBASE`, reside in the same lower 4GB aligned space. Bits [31:28] must be different between `CFGM_PERIPHBASE`, `DSU_PERIPHBASE`, and `DMC_PERIPHBASE`. Each of the three PERIPHBASE values are determined by the associated input strap signals.


Note

AxADDRU connected to DSU is 24 bits wide, however a DSU AXU interface can only address 1MB of address space.

The MSB AxADDRU[23] can be programmed to indicate whether the AXU request comes from APB vs CHI/AXI. See 4.3.1.4 por_axu_control on page 287

3.2 Clocks and resets

CMN-700 provides a hierarchical clocking microarchitecture which enables dynamic clock management for power efficiency. It also has a global reset signal.

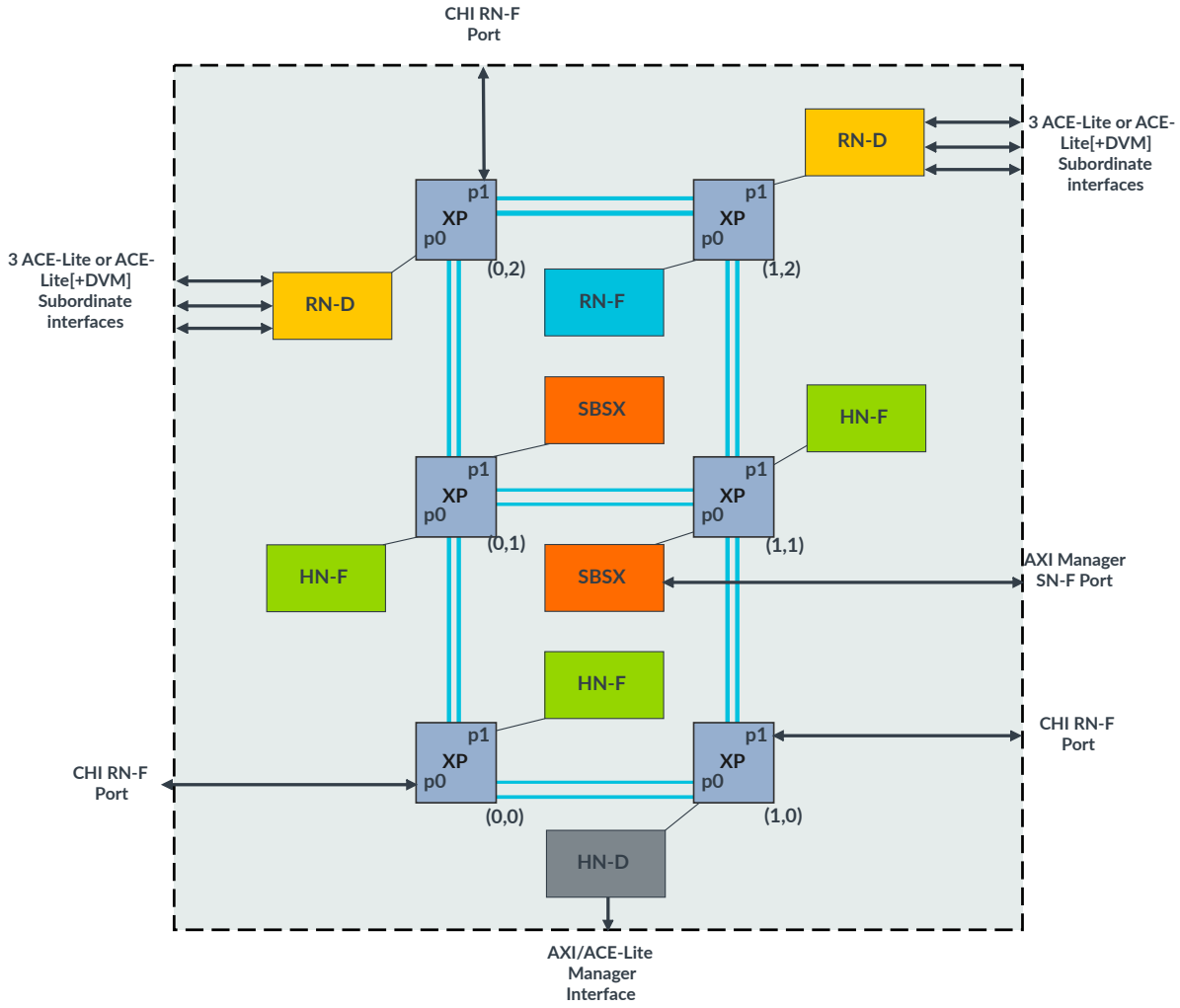
CMN-700 has a *High-level Clock Gating* (HCG) mechanism for clock signal management during periods of inactivity. See 3.2.5 High-level Clock Gating on page 84.

3.2.1 Clock domain configurations

CMN-700 supports either one synchronous or multiple asynchronous clock domains.

The following figure shows a CMN-700 configuration operating in a single and fully synchronous clock domain.

Figure 3-23: CMN-700 topology with fully synchronous clock domain

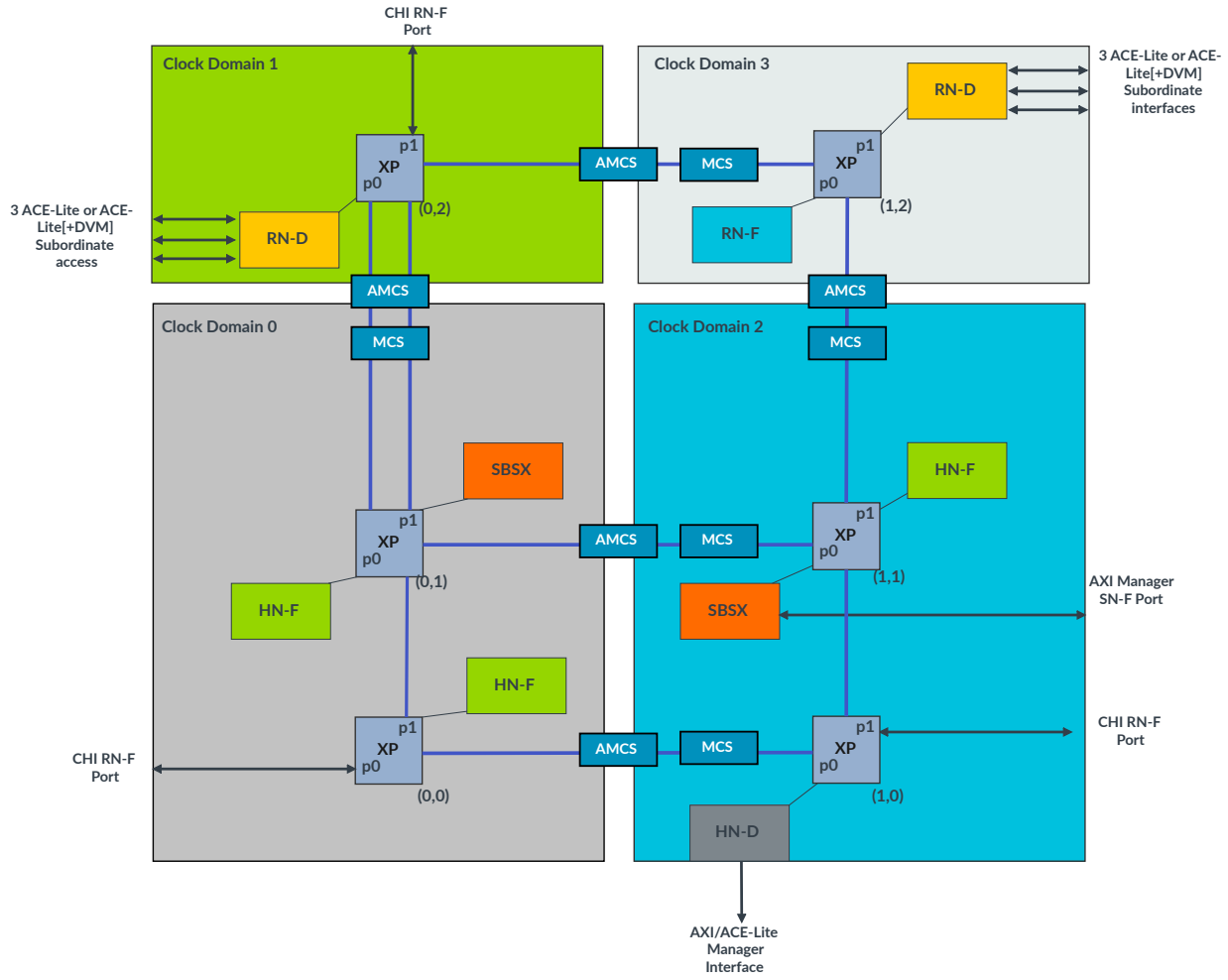


The global clock signal in a fully synchronous clock domain configuration is known as GCLK0.

For larger mesh topologies, synchronization and clock skew can be problematic because of the large distances that clock signals travel. Therefore, CMN-700 also supports dividing the mesh into four asynchronous clock domains. This feature is configured using Socrates™.

The following figure shows an example CMN-700 configuration with four asynchronous clock domains.

Figure 3-24: CMN-700 topology with four asynchronous clock domains



Each asynchronous clock domain is supplied by an individual clock input. These clock signals are known as GCLK0, GCLK1, GCLK2, and GCLK3, and collectively referred to as GCLKn.

If you configure the CMN-700 mesh to use multiple asynchronous clock domains, you must comply with the following restrictions:

- AMCSs must bridge asynchronous clock domains. Therefore, you must place AMCSs on the XP-XP links that span clock domains.
- The individual clock signals that supply each clock domain must run at the same frequency as each other, although they can be asynchronous.
- CMN-700 only supports rectangular clock domains containing one or more XPs. L-shaped or other clock domain topologies are not supported.
- 2 x 2 and smaller mesh configurations do not support multiple clock domains.

- A single Q-Channel controls all the clock signals. Therefore the clock domains cannot be gated separately, and must be powered up and down in the same way as a single synchronous clock domain.
- The mesh can only have a single power domain.

3.2.2 CCG clock inputs

GCLKn clocks the entire CCG logic, including its CXS interface. Optionally, if the external controller IP is running asynchronously to CMN clock (GCLKn), a CXS Domain Bridge (CXSDB) can be added on the CXS interface. The following figures show examples of this configuration type.

Figure 3-25: CMN-700 with CCG configuration clock domains with synchronous CXS domain

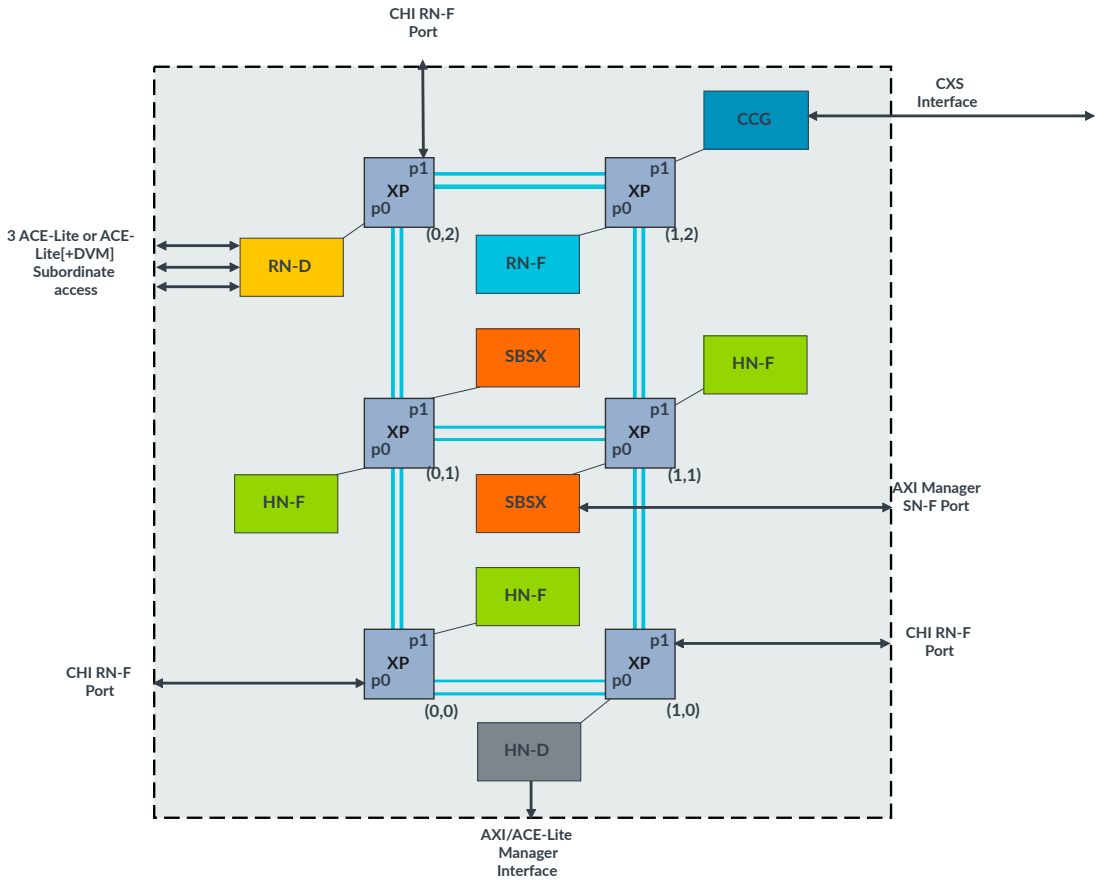
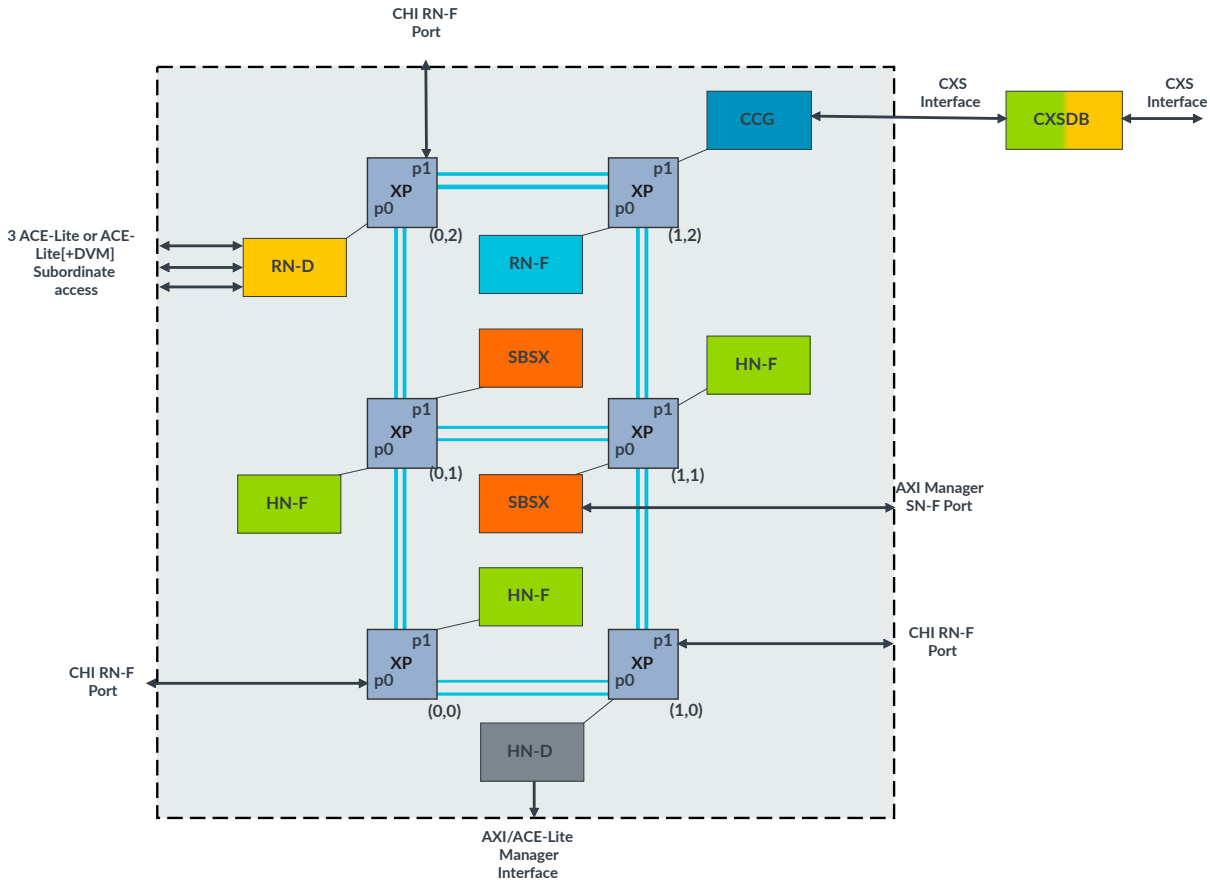


Figure 3-26: CMN-700 with CCG configuration clock domains with asynchronous CXS domain



3.2.3 Clock hierarchy

The clocking delivery and clock gating architecture are hierarchical.

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

Global clock The global clock is the clock input to the CMN-700 system. Another level of clock gating or clock control outside of the system is likely to control the global clock that is provided by the SoC. Although it is not a system requirement, CMN-700 includes support for external clock control.

Note

If you configure CMN-700 to use multiple asynchronous clock domains, a single Q-Channel controls each individual clock signal. Therefore, the individual clock domains cannot be separately gated. See 3.2.1 Clock domain configurations on page 78.

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. Regional clock gaters can shutdown the clock network between regional and local gaters. Therefore, this level of hierarchy enables greater power reduction than is possible using local clock gating. The regional clock gaters are instantiated in and controlled by the CMN-700 RTL. The exact set of regional clocks is internal to CMN-700 and is not described in this book.

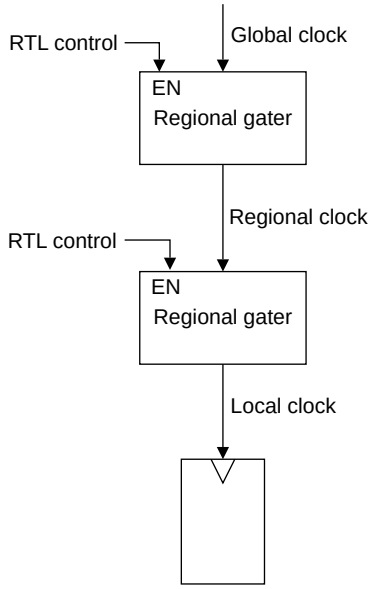
Local clocks Local clocks are created according to the following hierarchy:

1. RTL creates fine grained enable signals
2. Fine grained enable signals control local clock gaters
3. Local clock gaters output local clock signals

Local clock signals are used to directly clock sequential elements in CMN-700. The exact set of local clocks is internal to CMN-700 and is not described in this book.

The following figure shows the clocking hierarchy.

Figure 3-27: Clocking hierarchy



3.2.4 Clock enable inputs

CMN-700 includes several clock enable inputs.


Note

The following describes the relationship of the clock enable inputs to GCLK0. If your configuration uses multiple asynchronous clock domains, then the same relationship applies to the individual global clock signals for each clock domain.

The clock enable input signals are:

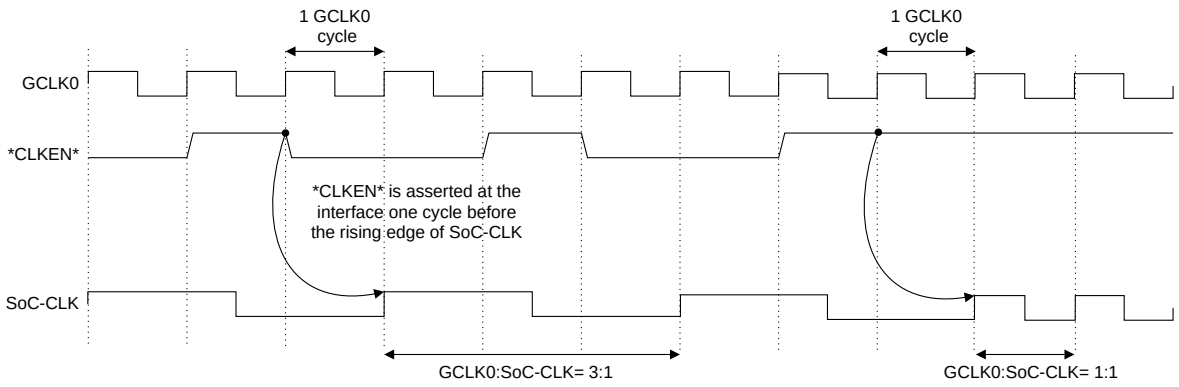
ACLKEN_S Present on each AMBA® subordinate interface.
ACLKEN_M Present on each AMBA® manager interface.
ATCLKEN Present on each debug and trace ATB interface.

All 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. The effect is that the clock enables run slower than GCLK0, ranging from ratios of 1:1 to 4:1. ATCLKEN is limited to 1:1, 2:1, and 4:1 integer fractions. This approach 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 a *CLKEN* ratio change. In the example, *CLKEN* changes the ratio of the relevant interface frequency relative to GCLK0 from 3:1 to 1:1.

Figure 3-28: *CLKEN* with GCLK0:SoC-CLK ratio changing from 3:1 to 1:1



3.2.5 High-level Clock Gating

The PCCB supports a *High-level Clock Gating* (HCG) mechanism. This mechanism notifies the SoC when CMN-700 is inactive and therefore reduces dynamic power consumption.

HCG enables an external SoC clock control unit, the *External Clock Controller* (ExtCC), to stop the GCLKn clock inputs. For more information about the ExtCC, see 3.2.6 External Clock Controller on page 85.


Note

If your CMN-700 configuration has multiple asynchronous clock domains, the ExtCC stops each individual global clock signal identically.

CMN-700 includes a Q-Channel interface that enables CMN-700 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*.

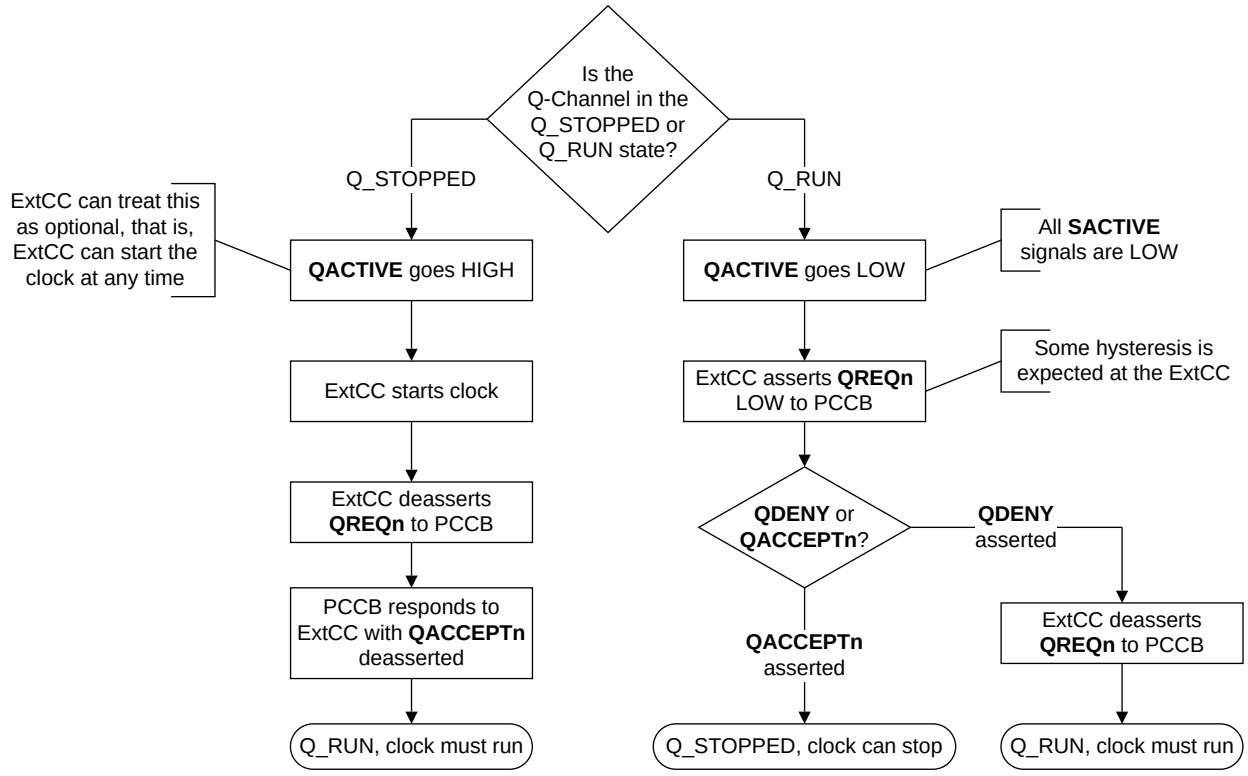
3.2.6 External Clock Controller

The ExtCC is used to control the clock gating flow.

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 3-29: Clock gating control using ExtCC



The requirements of the ExtCC are as follows:

- It must supply a clock to CMN-700 when the Q-Channel is in any state other than Q_STOPPED.
- The ExtCC can either choose to gate the clock to CMN-700 when the Q-Channel is in the Q_STOPPED state, or it can choose to run the clock at any time.
- ExtCC is responsible for bringing the Q-Channel to Q_RUN state after reset deassertion.
- Although this book 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. Therefore HCG is enabled when the system is inactive for long periods, but is not enabled for short periods of inactivity. If the clocks are stopped in response to short periods of inactivity, performance of CMN-700 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 CMN-700. More specifically, CMN-700 has no internal means of disabling HCG.

3.2.7 CCG clock management

CML CCG block runs at CMN-700 clock domain (GCLKn). It adds Q-Channel interface for each CCG instance.

CLK_CXS Q-Channel

Manages the CXS link interface.


Note

Optionally, CXS Domain Bridge (CXSDB) can be added on each of these CXS interfaces (IssueB) to operate CXS interface asynchronously to GCLKn. This CXSDB sits outside CCG block.

3.2.8 Reset

CMN-700 has a single global reset input signal, nSRESET.All paths from reset synchronizers are required to meet single cycle timing.

nSRESET is an active-LOW signal that can be asynchronously or synchronously asserted and deasserted.

When asserted, nSRESET must remain asserted for 90 clock cycles. Likewise, when deasserted, nSRESET must remain deasserted for 90 clock cycles. This requirement ensures that all internal CMN-700 components enter and exit their reset states correctly.

All CMN-700 clock inputs must be active during the required 90-cycle, or larger, period of nSRESET assertion. The clock inputs must also remain active for at least 90 cycles following deassertion of nSRESET.

3.3 Power management

CMN-700 includes several power management capabilities, that are either externally controllable or are assisted by the SoC.

CMN-700 has the following power management capabilities:

- Several distinct predefined power states. These states include ones in which all, half, or none of the SLC Tag and Data RAMs can be powered up, powered down, or in retention:
  - A state in which only the HN-F SF is active
  - A state in which the SLC RAMs and SF RAMs are inactive

  These power states reduce static and dynamic power consumption.
- Support for static retention in HN-F in which the SoC places SLC and SF RAMs in a retention state. This capability reduces static power consumption.

- Support for in-pipeline low-latency Data RAM retention control, in which a programmable idle counter can be used to put the SLC RAMs in retention.


Note

The clocking hierarchy and clock gating mechanism are described in 3.2 Clocks and resets on page 78.

3.3.1 Power domains

The power domains in CMN-700 are split between the logic and RAMs within the HN-F partitions.

The power domains are:

Logic

All logic except HN-F SLC Tag and Data RAMs and HN-F SF RAMs.

System Level Cache RAM0

SLC Tag and Data RAMs way[7:0] within HN-F partitions. The RAMs in each HN-F partition can be independently controlled.

System Level Cache RAM1

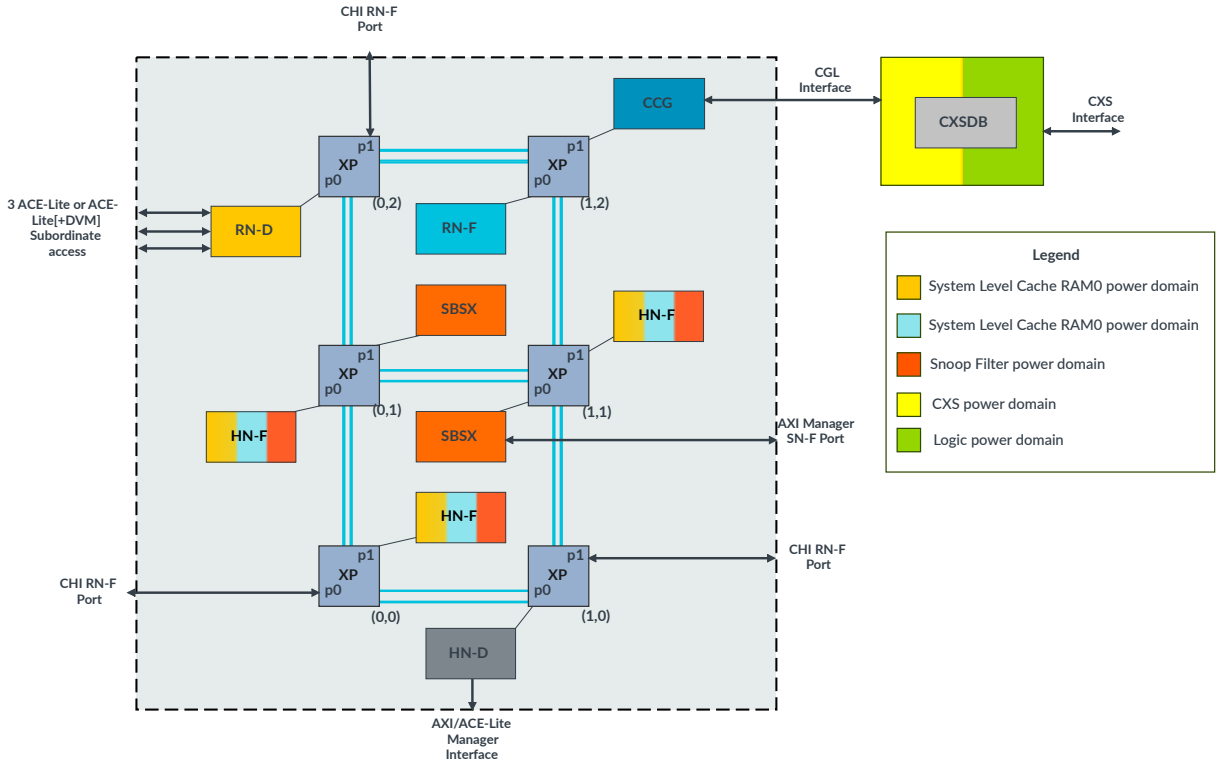
SLC Tag and Data RAMs way[15:8] within HN-F partitions. The RAMs in each HN-F partition can be independently controlled. The RAM1 domain for 384K, 1.5MB or 3MB SLC size configurations includes way[11:8].

Snoop filter only mode

SF RAMs within HN-F partitions. The RAMs in each HN-F partition can be independently controlled.

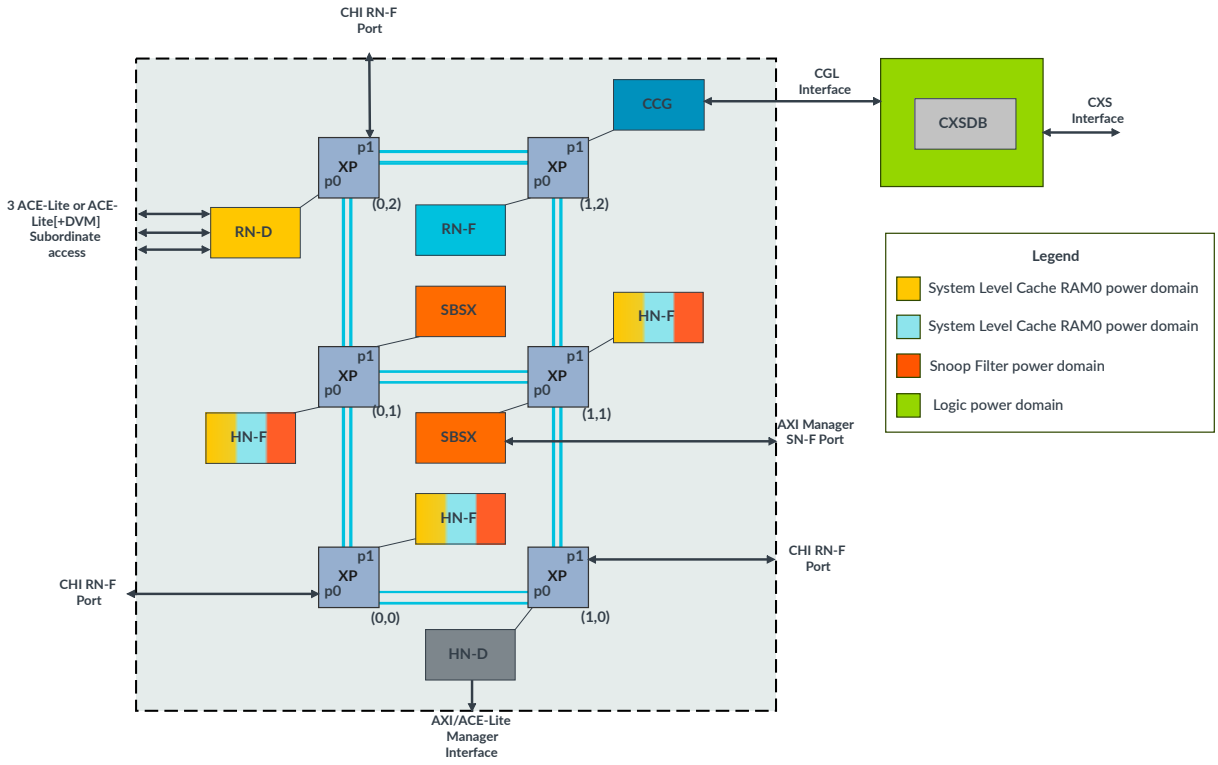
The following figure shows an example power domain configuration.

Figure 3-30: CMN-700 power domain example



The following figure shows another example power domain configuration, where the CXSDB component is in the same power domain.

Figure 3-31: Single CML power domain example

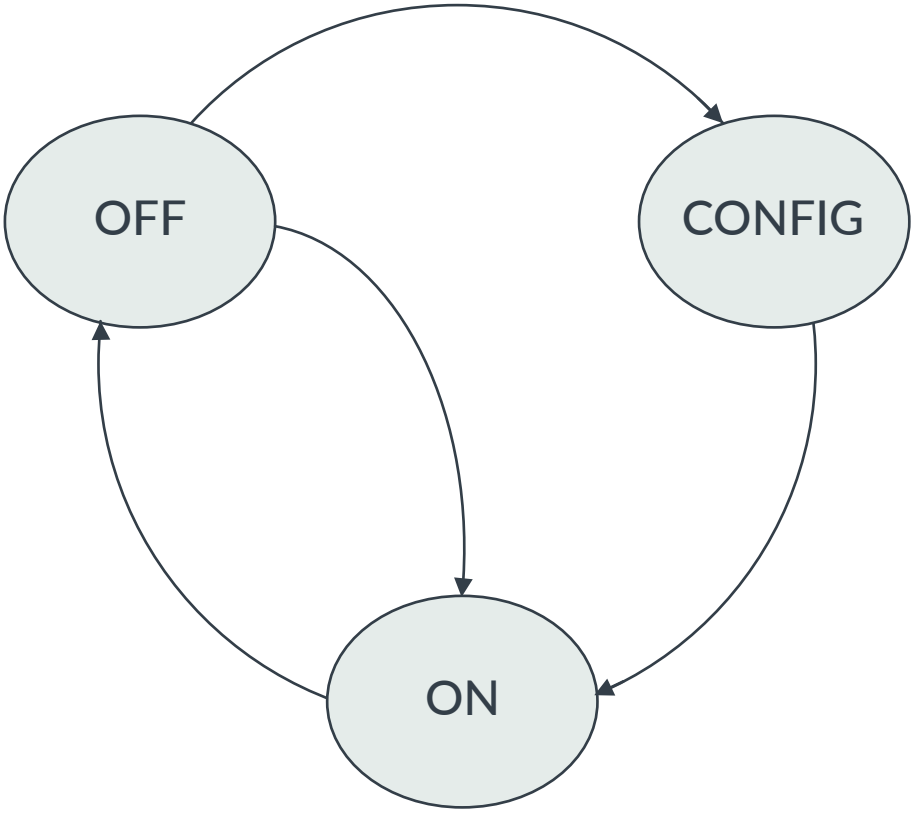


3.3.2 Power domain control

The CMN-700 Logic P-Channel controls all power domains except for the RAM and CXS power domains.

In addition to controlling the Logic domain, the Logic P-Channel allows synchronization between the HN-F software-controlled power domains and the Logic domain. This synchronization is achieved through a CONFIG state, as the following figure shows.

Figure 3-32: Logic domain states



There are two paths for transitioning from the OFF to ON state:

Cold reset

The Logic PSTATE OFF to ON transition also initiates NOSFSLC to FAM transition for all HN-F partitions.

Exit from HN-F Static Retention state

The Logic PSTATE transitions from OFF to CONFIG, indicating that CMN-700 is exiting a Memory Retention state, and does not initiate any HN-F partition power transitions.

The following table contains the power modes of components within the domain and the associated PSTATE values.

Table 3-8: Power mode configurations and PSTATE values

Power mode	PSTATE	CMN-700 logic	HN-F
OFF	0b00000	OFF	OFF/MEM_RET
CONFIG	0b11000	ON	ANY
ON	0b01000	ON	ANY

For an introduction to HN-F states, see 3.3.5 HN-F power domains on page 93.

For P-Channel signal list information, see B.14 Power management signals on page 1300.

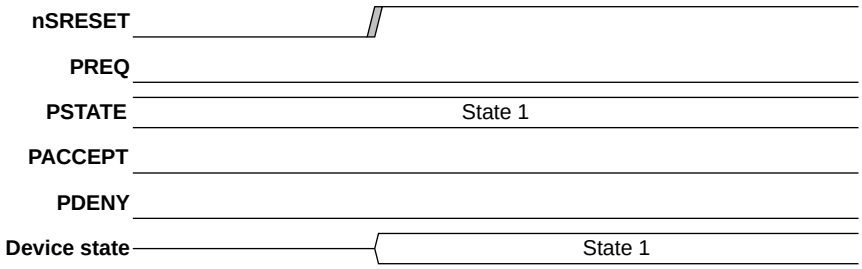
3.3.3 P-Channel on device reset

The following shows how to initialize the power state of a power domain.

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 nSRESET deasserts. The PSTATE inputs must be asserted before the deassertion of reset and remain after the deassertion of nSRESET, to allow reset propagation within CMN-700. The power controller must ensure that the reset sequence is complete before transitioning PSTATE, otherwise the device might sample an undetermined value. The following figure shows the state initialization on reset while PREQ is deasserted. CMN-700 also supports P-Channel initialization with PREQ asserted at nSRESET deassertion.

PSTATE inputs must be static 100 cycles before deassertion of nSRESET, and also for 100 cycles after the deassertion of nSRESET.

Figure 3-33: Reset state initialization with PREQ deasserted



For an introduction to HN-F states, see 3.3.5 HN-F power domains on page 93.

3.3.4 HN-F Memory retention mode

When isolating the CMN-700 outputs, handshake protocols on certain interfaces must be followed. The steps list how to enter and exit HN-F Memory retention mode.

Entering HN-F Memory retention mode

1. Program the HN-Fs to enter the required power state.
2. Quiesce the interconnect, and wait for QACTIVE to drop.
3. Place CMN-700 in LOGIC_OFF state through the Logic P-Channel.
4. Isolate the CMN-700 outputs. If the logic on the other side of the interface is being powered down or reset, this step might not be required.
5. Turn off power to CMN-700

Exiting HN-F Memory retention mode

1. Apply power to CMN-700
2. Assert reset
3. Enable clocks
4. Disable isolation of the CMN-700 outputs
5. Deassert reset
6. Place CMN-700 in LOGIC_CONFIG state through the logic P-Channel.
7. Reprogram the HN-F PWPR to the retention mode the HN-F was in before turning off power.
8. Reprogram the HN-F PWPR to ON
9. Reprogram the CMN-700 configuration registers, including the RN SAM and any other registers written during cold boot.
10. Place CMN-700 in LOGIC_ON state through the P-Channel.
11. Resume traffic/normal operation

3.3.5 HN-F power domains

The HN-F has various power states. Transitioning between different states enables or disables different parts of the HN-F.

The HN-F has three classes of power states:

1. Operational states, where logic is on and enabled RAMs are operating as normal
2. Functional retention states, where logic is on, and enabled RAMs are in retention
3. Memory Retention states, where logic is off and enabled RAMs are in retention

Within these power states, the HN-Fs in an SCG operates in four modes:

FAM

Full Associativity Mode (FAM), where the SF and the entire SLC are enabled

HAM

Half-Associativity Mode (HAM), where the SF is enabled but the upper half of the SLC ways are disabled and powered off

SFONLY

Snoop filter only mode (SFONLY), where the SF is enabled but the whole SLC is powered off

NOSFSLC

No-SLC Mode (NOSFSLC), where the SF and SLC are disabled and powered off

The following constraints apply to the power states and transitions:

- If SLC size is 0KB, the HN-F does not support transitions to FAM or HAM modes
- After initialization, the power status register indicates FAM instead of SFONLY for 0KB SLC configurations

- When a power transition is initiated, another must not be initiated until the first one completes

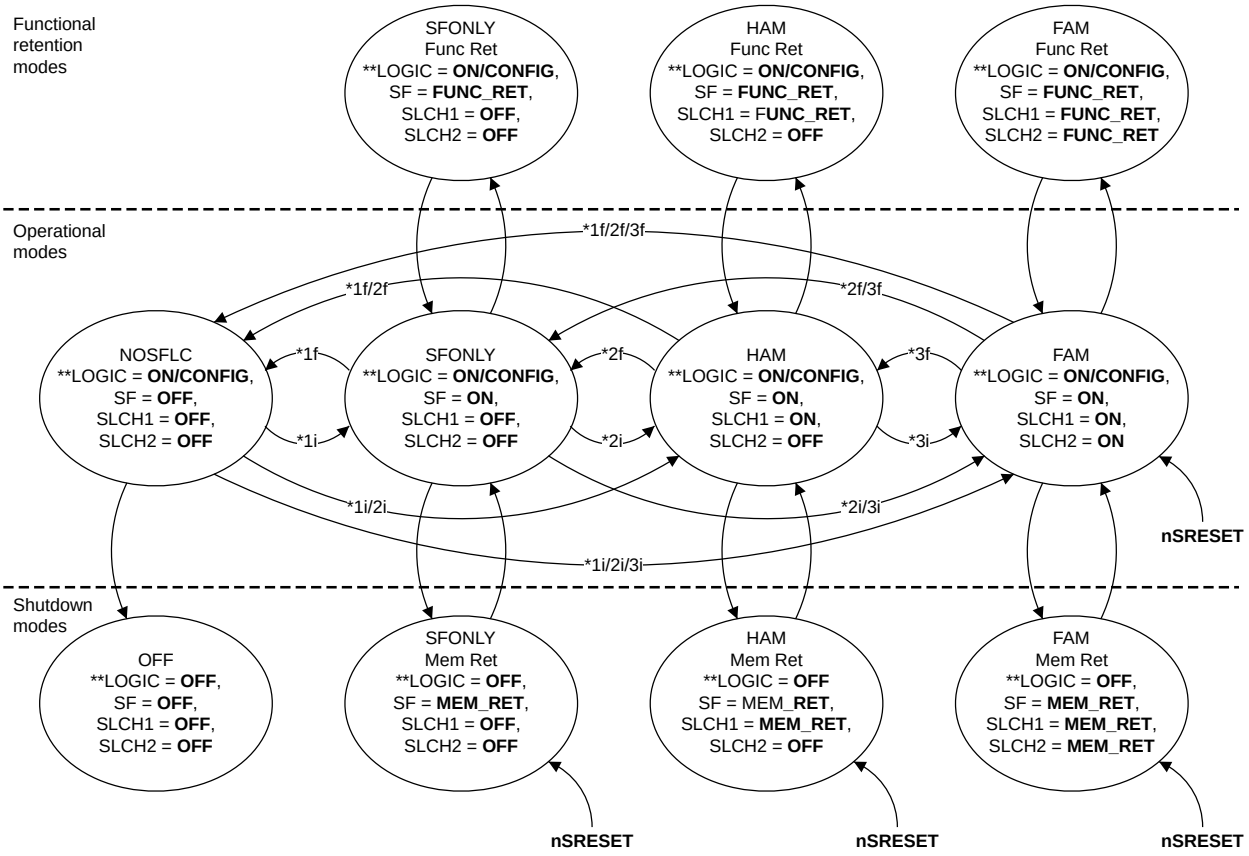
The following table shows the valid HN-F power states and their requirements.

Table 3-9: HN-F power states

State	Description	Control logic	SF power state	SLC way[7:0] power state	SLC way[15:8] power state
FAM	Full Run mode	On	On	On	On
HAM	Run mode with SLCH2 (SLC upper ways) disabled	On	On	On	Off
SF	Run mode with SLCH1 and SLCH2 disabled	On	On	Off	Off
NOSFSLC	Run mode with SLCH1, SLCH2, and SF disabled	On	Off	Off	Off
FAM FUNC_RET	Run mode with SLCH1, SLCH2, and SF in dynamic retention	On	Retention	Retention	Retention
HAM FUNC_RET	Run mode with SLCH1 and SF in retention, and SLCH2 in power down	On	Retention	Retention	Off
SF FUNC_RET	Run mode with SF in retention, and SLCH1 and SLCH2 in power down	On	Retention	Off	Off
FAM MEM_RET	Run mode with SLCH1 and SF in retention, and SLCH2Shut down with SLCH1, SLCH2, and SF in retention	Off	Retention	Retention	Retention
HAM MEM_RET	Shut down with SLCH1 and SF in retention, and SLCH2 in power down	Off	Retention	Retention	Off
SF MEM_RET	Shut down with SF in retention, and SLCH1 and SLCH2 in power down	Off	Retention	Off	Off

The following figure shows the valid power states and transitions for a CMN-700 system.

Figure 3-34: Power state transitions



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.

The SF does not track RN-F coherence while the HN-F is in NOSFSLC state. Therefore, RN-Fs must be quiesced down before the flush because of power state transition to NOSFSLC. The RN-F caches must be flushed before transitioning from NOSFSLC to SFONLY, HAM, or FAM states.

These HN-F power states are transitioned using configuration register writes that must target all HN-Fs in the SCG region. Also, the logic domain P-Channel interface can initiate a NOSFSLC→FAM transition.


Note

CMN-700 does not accept requests before SLC initialization has completed.

Write to the following cmn_hns_ppu_pwpr register fields to transition HN-F partitions to a required power state:

- Policy
- op_mode

When the power state transition is complete, the following cmn_hns_ppu_pwsr register fields are updated:

- pow_status
- op_mode_status

If either the SLC, SF, or both are flushed as part of a power transition, then the power state transition can take thousands of clock cycles. Also, the INTREQPPU interrupt output can be used to indicate the completion of the HN-F power state transitions.

From the FAM, HAM, or SFONLY modes, the HN-F can enter a dynamic retention mode using configuration register writes, 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
- The array pipeline is blocked, and a handshake occurs to allow array access when exiting the retention state

These dynamic power transitions are executed autonomously within each HN-F partition. Each HN-F has a programmable idle cycle counter and initiates a P-Channel handshake with the corresponding RAMs to enter the dynamic retention state. The pipeline then blocks transactions that target the HN-F RAMs. A coherent transaction triggers an exit from the dynamic retention state, initiates another P-Channel handshake, and takes the RAMs out of dynamic retention mode.

From these states, the SLC can also enter a Memory 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
- Reset deassertion is essential when exiting retention after logic power down

A P-Channel interface controls the CMN-700 logic domain power state. This P-Channel interface also interacts with the HN-F power control logic using an internal bus. The HN-F power control logic waits for a command on the deassertion of nSRESET, depending on the overall power state transition that is required. For the Cold reset HN-F FAM transition case, the PCCB block initiates the HN-F NOSFSLC→FAM command. To exit static retention cases, the SCP initiates configuration register writes to the HN-F to indicate the HN-F power state.

The circumstances where the HN-F enters dynamic retention modes or static retention modes are different. Dynamic retention is entered because of a dynamic activity or inactivity indicator from the HN-F to the SoC. This indicator is an output of the HN-F, and is used to determine periods of inactivity long enough to warrant entering retention mode. However the inactivity is either not long

enough or not the type of inactivity to make the SoC place the SLC and SF into static retention. In addition to the static retention modes, the control logic can be powered down from the NOSFSLC state, at which point CMN-700 is fully off.

The HN-Fs performs all activity that is required to enable safe transition between the respective power states automatically in response to input P-Channel PSTATE transitions. It is not necessary for the SoC logic to perform any additional activity to enable transitions between power states. For example, the HN-F performs clean and invalidation of half of the ways of the SLC and clean and invalidation of all ways of the SLC. This clean and invalidation activity occurs requires the respective power state transitions.

CMN-700 cannot make any power transitions while the control logic is powered off. Consider a transition from FAM static retention to OFF. To complete this transition, the power state must first move through FAM, and NOSFSLC states while the LOGIC power domain is on. These transitions allow the SLC and SF to be flushed.

The following table shows the PSTATE encodings for the HN-F and power domains including RAM configurations for the different operational modes.

HN-F cannot process any transactions while in static retention (FUNC_RET or MEM_RET). HN-F must be in the ON state before sending any transactions to HN-F in this case. If HN-F is in dynamic retention, any activity autonomously takes HN-F out of dynamic retention.

Table 3-10: Power modes, operational modes, and RAM configurations

Operational mode	Power mode	PSTATE	Bank 0 RAM	Bank 1 RAM	SF RAM
FAM	ON	11_1000	ON	ON	ON
	FUNC_RET	11_0111	RET	RET	RET
	MEM_RET	11_0010	RET	RET	RET
HAM	ON	10_1000	ON	OFF	ON
	FUNC_RET	10_0111	RET	OFF	RET
	MEM_RET	10_0010	RET	OFF	RET
SFONLY	ON	01_1000	OFF	OFF	ON
	FUNC_RET	01_0111	OFF	OFF	RET
	MEM_RET	01_0010	OFF	OFF	RET
NOSFSLC	MEM_OFF	00_0110	OFF	OFF	OFF
	OFF	00_0000	OFF	OFF	OFF

3.3.6 HN-F RAM PCSM Interface

Each HN-F RAM interface contains a *Power Control State Machine* (PCSM).

Each PCSM P-Channel interface that can be used to convert power state transitions into technology-specific controls, and the overall HN-F partition power state transition, depends on all P-Channel transactions to complete.

The following table lists the valid PSTATE values for this interface.

Table 3-11: PSTATE Encodings

PSTATE	Value
ON	0b1000
FUNC_RET	0b0111
MEM_RET	0b0010
OFF	0b0000


Note

This interface does not have a PDENY signal.

3.3.7 HN-F power domain completion interrupt

The PCCB can be configured to generate interrupt INTREQPPU on completion of power state transitions for a collection of HN-Fs.

The PCCB contains a global status register, por_ppu_int_status, which indicates HN-F power state transition completion. The PCCB also contains a mask register, por_ppu_int_mask, which allows filtering on all or a subset of the HN-Fs in the CMN-700 configuration. The bit positions in the por_ppu_int_status and por_ppu_int_mask registers correspond to the logical ID of the HN-Fs.

INTREQPPU asserts when all por_ppu_int_status register bits with the corresponding por_ppu_int_mask register bit are set by the HN-F power transition completion.

To deassert INTREQPPU, write `0b1` to the bits of the por_ppu_int_status register that correspond to the masked group of HN-Fs that completed the power transitions.

3.3.8 RN entry to and exit from Snoop and DVM domains

CMN-700 includes a feature that allows RNs to be included or excluded from the system coherency domain. This domain is also known as the Snoop domain or DVM domain. This feature ensures correct operations of Snoops and DVMs when:

- An RN is taken out of reset.

- An RN is powered down and then later powered up.

RN-Fs behave as follows:

- If an RN-F is included in the system coherency domain, it must respond to Snoop and DVM requests from CMN-700.
- If an RN-F is excluded from the system coherency domain, it does not receive Snoop or DVM requests from CMN-700.

RN-Ds behave as follows:

- If an RN-D is included in the system coherency domain, it must respond to DVM requests from CMN-700.
- If an RN-D is excluded from the system coherency domain, it does not receive DVM requests from CMN-700.

3.3.8.1 Hardware interface

The following describes the hardware interface for RN inclusion into and exclusion from system coherency domain.

CMN-700 provides two signals for RN system coherency entry and exit:

- SYSCOREQ input (to CMN-700)
- SYSCOACK output (from CMN-700)

These two signals implement a four-phase handshake between the RN and CMN-700 with the four states as specified in the following table.

Table 3-12: RN system coherency states

SYSCOREQ	SYSCOACK	State
0	0	DISABLED
1	0	CONNECT
1	1	ENABLED
0	1	DISCONNECT

Coming out of Reset, the RN is in the DISABLED system coherency state.

CONNECT

To enter system coherency, RN must assert SYSCOREQ and transition to CONNECT state. The RN must be ready to receive and respond to Snoop and DVM requests in this state.

ENABLED

Next, CMN-700 asserts SYSCOACK and transitions to ENABLED state. The RN is now included in the system coherency domain. The RN can receive and must respond to Snoop and DVM requests in this state.

DISCONNECT

When the RN is ready to exit system coherency, it must deassert SYSCOREQ and transition to DISCONNECT state. The RN continues to receive and must respond to Snoop and DVM requests in this state.

DISABLED

When all outstanding Snoop and DVM responses have been received, CMN-700 deasserts SYSCOACK and transitions to DISABLED state. In this state, no snoop or DVM transactions are sent to the RN which can now be powered down.

The following rules must be obeyed to adhere to the four-phase handshake protocol.

- When SYSCOREQ is asserted, it must remain asserted until SYSCOACK is asserted.
- When SYSCOREQ is deasserted, it must remain deasserted until SYSCOACK is deasserted.

3.3.8.2 Software interface

The following describes the software interface for RN inclusion into and exclusion from system coherency domain.

CMN-700 provides two *Configuration Registers* (CRs) for system coherency entry and exit:

RN-F 4.3.13.31 por_mxp_p0-5_syscoreq_ctl on page 918
RN-D 4.3.14.15 por_rnd_syscoreq_ctl on page 987

Reading and writing to these CRs provides a software alternative to the 4-phase hardware handshake.

It is possible the CRs contain multiple bits where each bit corresponds to a different RN. The following description is about the read and write of the CR bit that corresponds to a given RN. When configuring the system coherency entry or exit for a given RN, software must adopt a Read-Modify-Write strategy. This strategy ensures CR bits corresponding to other RNs, are not modified when writing into the syscoreq_ctl CR.

Coming out of reset, both CRs are cleared, indicating DISABLED state.

CONNECT

To initiate an RN entry into the system coherency domain, software must first poll both CRs. This poll ensures the CR bits corresponding to that RN, are set to 0. When the RN is ready to receive and respond to Snoop and DVM requests, software must write 1 into the corresponding bit in the syscoreq_ctl CR. This write process transitions the RN to CONNECT state.

ENABLED

Next, CMN-700 indicates a transition to ENABLED state by setting the corresponding CR bit in the syscoack_status register to 1. The RN is now inside the system coherency domain. Software can poll the syscoack_status register to determine this state transition.

DISCONNECT

To initiate an RN exit from the system coherency domain, software must first poll both CRs. After ensuring that the CR bits corresponding to that RN are set, software must clear the corresponding syscoreq_ctl bit to transition the RN to DISCONNECT state. The RN continues to receive and must respond to Snoop and DVM requests in this state.

DISABLED

When all outstanding Snoop and DVM responses have been received, CMN-700 clears the corresponding syscoack_status bit indicating the transition to DISABLED state. In this state, no Snoop or DVM transactions are sent to the RN. Software must poll the syscoack_status register to ensure that this state transition has occurred before initiating RN powerdown.


Note

To adhere to the four-phase handshake protocol, the following rules apply:

- The hardware interface is intended as the primary interface. The software interface is provided as an alternative for legacy RN devices and systems that do not support the hardware interface.
- Either hardware or software interface must be used, but not both. Coming out of reset, the hardware interface is enabled by default. The first write into the syscoreq_ctl register disables hardware interface and enables software interface. There must be a reset to re-enable hardware interface.
- When software interface is employed, SYSCOREQ must remain deasserted.
- When hardware interface is employed, software must not write to the syscoreq_ctl CR.

3.4 Network layer functions

CMN-700 has specific functions that it uses to map regions of the address space, determine flit targets, and define overall routing behavior. Some of these functions are configurable by setting configuration parameters or by software.

3.4.1 Node ID mapping

The physical position of a device in the mesh determines the node ID mapping.

This scheme is the default node ID-mapping scheme. This scheme is different from the one that is used when using extra device ports on MXPs. For more information about that scheme, see 3.4.2 Node ID mapping for configurations with extra device ports on page 106.

The following details determine the physical position of a device in the mesh:

1. The X coordinate of its XP
2. The Y coordinate of its XP
3. The XP device port (0 or 1) that it connects to

The device node ID is mapped to (X, Y, Port, `0b00`).

1. The bit widths of the X and Y parameters depend on the configured size of the mesh.
2. The naming convention for I/O signals uses decimal values of the node ID. For example, `RXREQFLIT_NIDxxx` uses **`xxx`** values in decimal.

The node ID size depends on the X and Y dimensions of the CMN-700 mesh. The larger of the X and Y dimensions determines the size, as the following table shows.

Table 3-13: Node ID size selection

X mesh dimension	Y mesh dimension	Node ID size
4 or less	4 or less	7 bits
5-8	8 or less	9 bits
8 or less	5-8	
9 or more	9 or more	11 bits

On internal CHI interfaces, the NodeID width is set equal to the NodeID size. On external CHI interfaces, the NodeID width is set to 11 bits, where unused bits occupy MSBs and are driven to zeroes.

The following tables contain the different node ID formats.

Table 3-14: 7-bit node ID format

[6:5]	[4:3]	[2]	[1:0]
X position	Y position	Port	`0b00`

Table 3-15: 9-bit node ID format

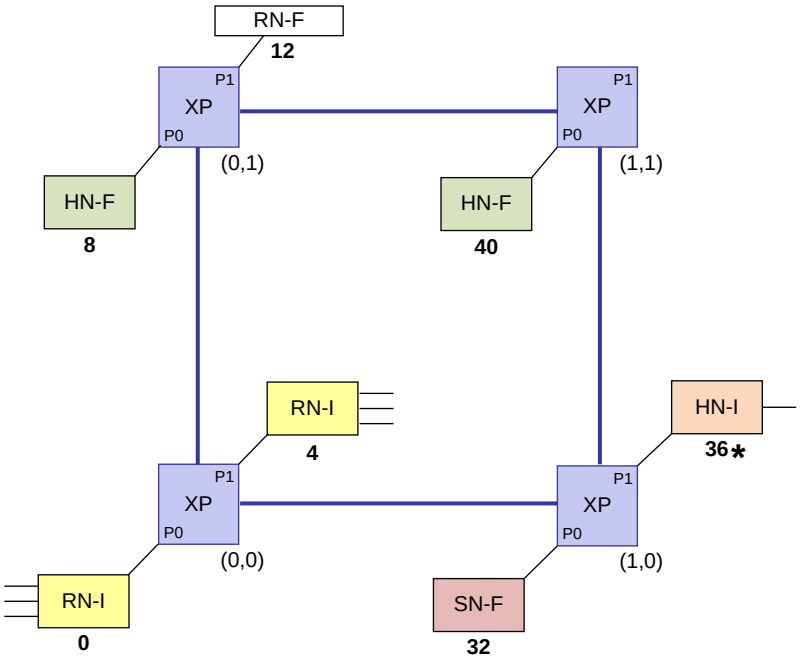
[8:6]	[5:3]	[2]	[1:0]
X position	Y position	Port	`0b00`

Table 3-16: 11-bit node ID format

[10:7]	[6:3]	[2]	[1:0]
X position	Y position	Port	`0b00`

The following figure shows a CMN-700 system with 7-bit node IDs in (X, Y, Port, DeviceID) format.

Figure 3-35: Example system with 7-bit node IDs



Example 3-2: 7-bit node ID format

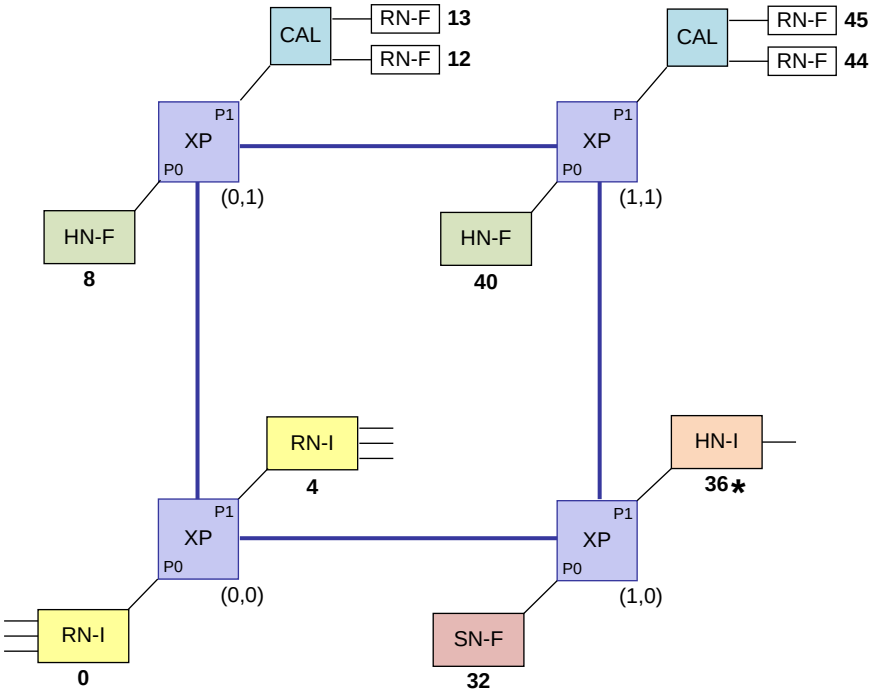
For the HN-I connected to XP (1,0), the node ID reads as (1,0,1,0).

This format is equivalent to (`0b01`, `0b00`, `0b1`, `0b00`) or `0x24`.

If CAL is present, the two devices that are connected to the CAL are assigned consecutive node IDs. One device is assigned NodeID[1:0]=`0b00` and the other device is assigned NodeID[1:0]=`0b01`.

The following figure shows a CMN-700 system with CAL and 7-bit node IDs in (X, Y, Port, DeviceID) format.

Figure 3-36: Example system with 7-bit node IDs and CAL

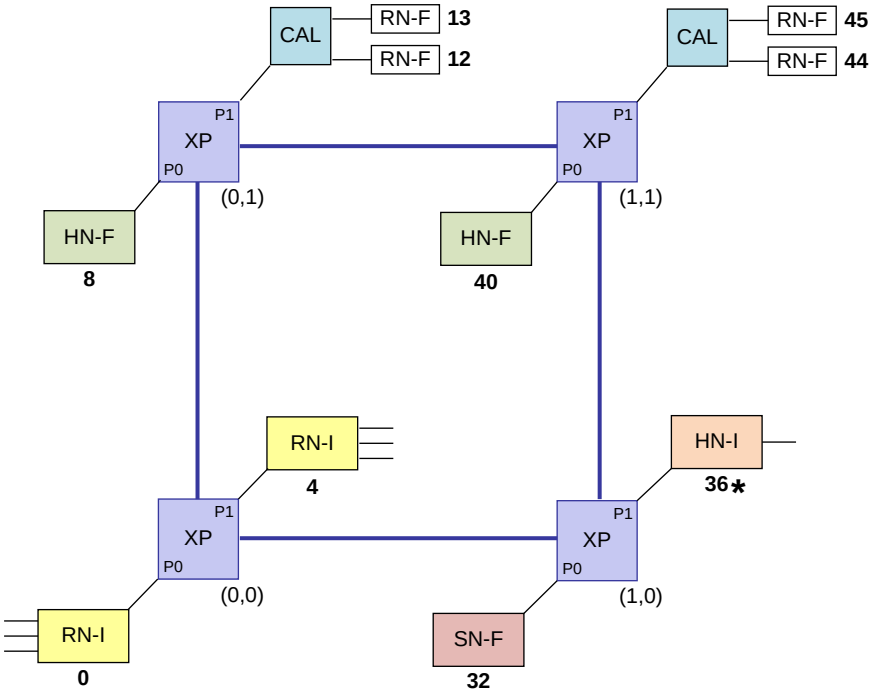


If CAL is present, the devices that are connected to the CAL are assigned consecutive node IDs. For a CAL2, one device is assigned NodeID[1:0]=`0b00` and the other device is assigned NodeID[1:0]=`0b01`. For a CAL4:

- Device 0 NodeID[1:0] = `0b00`
- Device 1 NodeID[1:0] = `0b01`
- Device 2 NodeID[1:0] = `0b10`
- Device 3 NodeID[1:0] = `0b11`

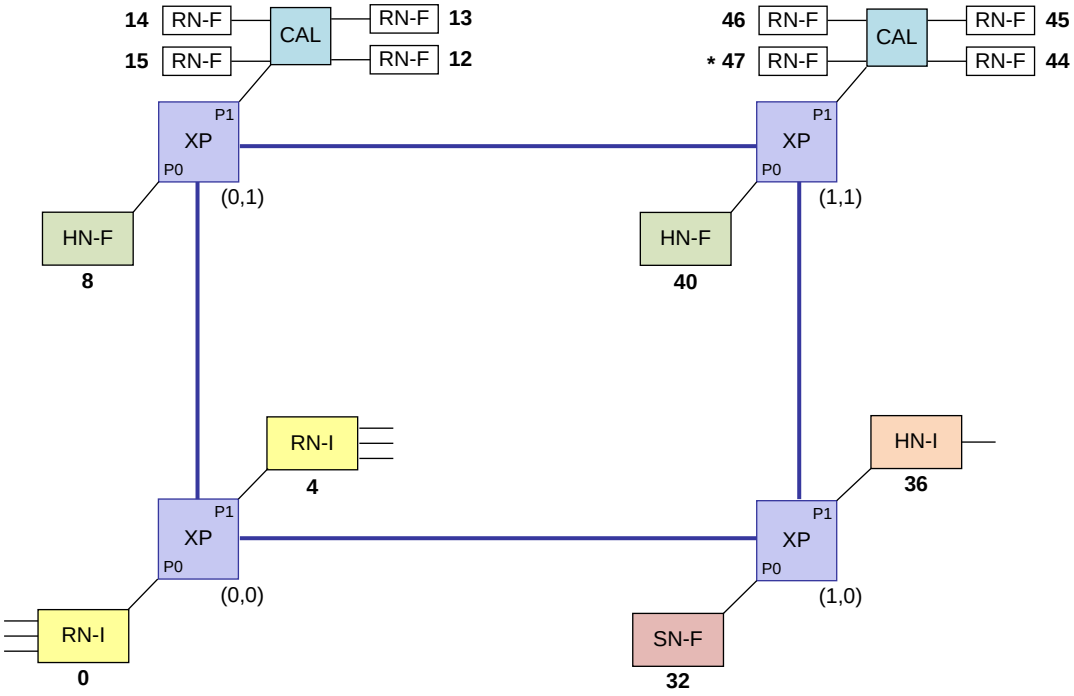
The following figure shows a CMN-700 system with CAL2 and 7-bit node IDs in (X, Y, Port, DeviceID) format.

Figure 3-37: Example system with 7-bit node IDs and CAL2



The following figure shows a CMN-700 system with CAL4 and 7-bit node IDs in (X, Y, Port, DeviceID) format.

Figure 3-38: Example system with 7-bit node IDs and CAL4



Example 3-3: 7-bit node ID format with CAL4

For the third RN-F connected to the CAL on XP (1,1), the node ID reads as (1,1,1,3).

This format is equivalent to `0b01`, `0b01`, `0b1`, `0b11`, or `0x2F`.

3.4.2 Node ID mapping for configurations with extra device ports

The node ID mapping for a CMN-700 configuration changes from the default scheme when you use extra device ports. In this case, the way that node IDs are mapped depends on whether you are using a mesh configuration.

The node ID-mapping scheme which is described here is used for configurations with extra device ports. This scheme is different to the default scheme. If your configuration does not use extra device ports, see 3.4.1 Node ID mapping on page 101.

CHI device node ID for mesh configurations

The naming convention for I/O signals uses decimal values of the node ID. For example, `RXREQFLIT_NIDxxx` uses **xxx** values in decimal.

The following items determine the physical position of a device in a CMN-700 mesh configuration with extra device ports:

1. The X coordinate of its XP
2. The Y coordinate of its XP
3. The XP device port (0, 1, 2, or 3) to which it is connected

The CHI device node ID is mapped to (X, Y, Device port plus device ID).

The node ID size depends on the X and Y dimensions of the CMN-700 mesh. The larger of the X and Y dimensions determines the size as the following table shows.

Table 3-17: Node ID size selection

X mesh dimension	Y mesh dimension	Node ID size
4 or less	4 or less	7 bits
5-8	8 or less	9 bits
8 or less	5-8	
9 or more	9 or more	11 bits

The following tables show the 7-bit and 9-bit node ID format for a CMN-700 mesh configuration using extra device ports.

Table 3-18: 7-bit node ID format for mesh configuration with extra device ports

[6:5]	[4:3]	[2:0]
X position	Y position	Device port plus device ID **[2:1]** Indicates device port **[0]** Indicates device ID on the device port

Table 3-19: 9-bit node ID format for mesh configuration with extra device ports

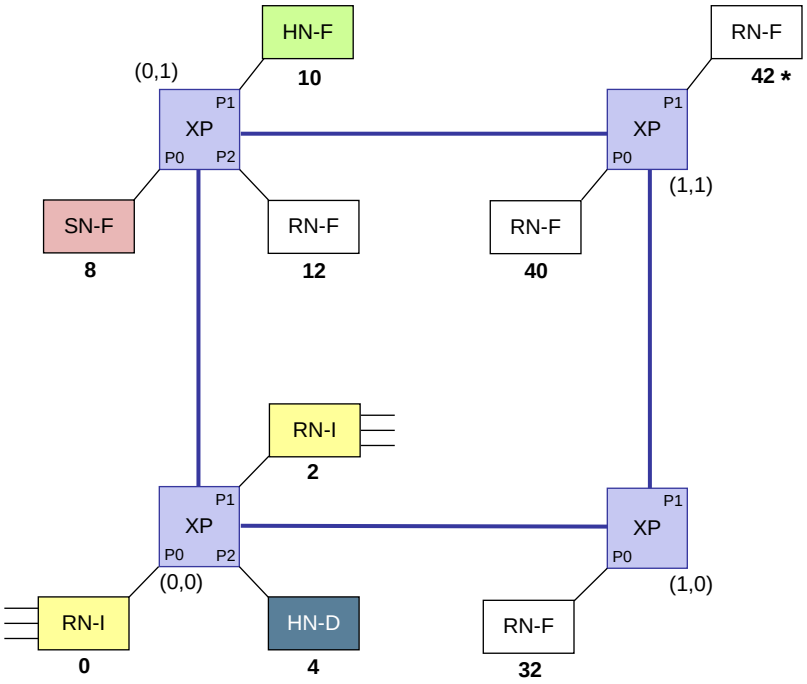
[8:6]	[5:3]	[2:0]
X position	Y position	Device port plus device ID **[2:1]** Indicates device port **[0]** Indicates device ID on the device port

Table 3-20: 11-bit node ID format for mesh configuration with extra device ports

[10:7]	[6:3]	[2:0]
X position	Y position	Device port plus device ID **[2:1]** Indicates device port **[0]** Indicates device ID on the device port

The following figure shows a CMN-700 system with node IDs in (X, Y, Device port plus device ID) format.

Figure 3-39: Example mesh system with node IDs



Example 3-4: Node ID format for mesh system with extra device ports

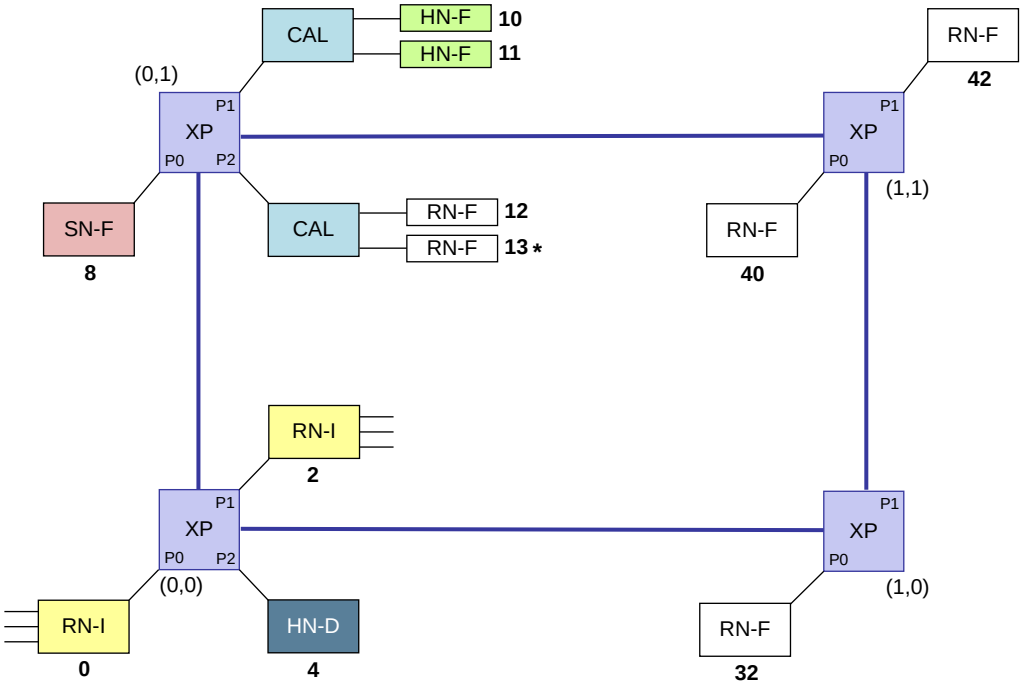
For the RN-F on P1 of XP (1,1), the node ID reads as (1,1,2).

This format is equivalent to (`0b01`, `0b01`, `0b010`) or `0x2A`.

If using CAL in a CMN-700 mesh configuration with extra device ports, only CAL2 configurations are permitted. If CAL is present, the devices that are connected to the CAL are assigned consecutive node IDs. For example, consider two devices that are attached to a CAL on P0. One device is assigned NodeID[2:0]=`0b000` and the other device is assigned NodeID[2:0]=`0b001`.

The following figure shows a CMN-700 system with CAL and node IDs in (X, Y, Device port plus device ID) format.

Figure 3-40: Example mesh system using CAL with node IDs



Example 3-5: Node ID format for mesh system with extra device ports and CAL

For the second RN-F attached to the CAL on P2 of XP (0,1), the node ID reads as (0,1,5).

This format is equivalent to (`0b00`, `0b01`, `0b101`) or `0xD`.

The following table shows the possible node ID[2:0] encodings for a mesh configuration with extra device ports.

Table 3-21: Node ID[2:0] encodings for mesh configuration with extra device ports

Device port number	Device number	Node ID[2:1]	Node ID[0]
P0	0	`0b00`	`0b0`
	1	`0b00`	`0b1`
P1	0	`0b01`	`0b0`
	1	`0b01`	`0b1`
P2	0	`0b10`	`0b0`
	1	`0b10`	`0b1`
P3	0	`0b11`	`0b0`
	1	`0b11`	`0b1`

3.4.3 Addressing capabilities

CMN-700 supports a 34-bit, 44-bit, 48-bit, or 52-bit *Physical Address* (PA) width which defines the PA space for which read and write transactions are supported in the interconnect. The PA width is configured using Socrates and results in the `PA_WIDTH` global parameter in the CMN-700 RTL.

CHI interfaces in CMN-700 support 44-bit, 48-bit, and 52-bit address field widths for REQ channel flits. This width is also configured using Socrates™ and results in the `REQ_ADDR_WIDTH` global parameter in the CMN-700 RTL.

The address field width for SNP channel flits is derived automatically based on the `REQ_ADDR_WIDTH` global parameter.

The following table shows the legal combinations of physical address widths and flit address widths.

Table 3-22: Legal combinations of physical address and flit address widths

Physical address width	REQ flit address width	SNP flit address width (derived)
34b	44b	41b
34b	48b	45b
44b	44b	41b
44b	48b	45b
48b	48b	45b
52b	52b	49b

3.4.4 System Address Map

Every manager that is connected to CMN-700 has the same view of memory.

The entire addressable space can be partitioned into subregions, and each partition must be designated as one of the following:

I/O space

HN-I, HN-D, HN-P, HN-T, and HN-V service requests to I/O space.

DDR space

HN-F, SN-F, and SBSX service requests to DDR space.


Note

Unmapped addresses are routed to the HN-D.

Each HN-F covers a mutually exclusive portion of the system address space. The options and constraints for HN-Fs are:

- Each HN-F can contain an SLC

- HN-Fs can be combined into *System Cache Groups* (SCGs)
- Each HN-F in an SCG must have the same SLC partition size
- An address hash function determines the target HN-F within an SCG

All CHI transactions require a target ID to route packets from source to destination. For addressable requests, a *System Address Map* (SAM) determines the target ID. Each node that can generate a CHI addressable request contains a SAM:

RN SAM

Present for all RNs and CML HA nodes. Generates a target ID for requests to HN-F, HN-I, HN-D, HN-P, HN-T, HN-V, SBSX, and SN-F.

CML RA SAM

Present in all CML RA nodes. Generates a target ID for requests to remote CML HA nodes.

HN-F SAM

Present in all HN-Fs. Generates a target ID for requests to SN-F and SBSX.

HN-I SAM

Present in all HN-Is. Maps the address of the incoming CHI request to an I/O subregion for ordering purposes.

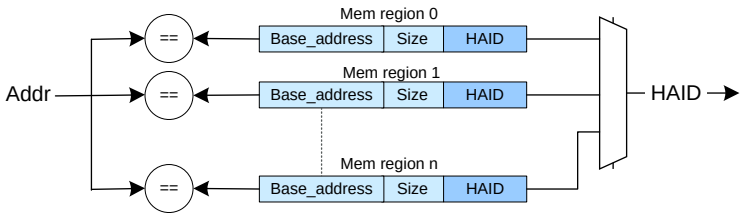
3.4.5 CML RA SAM

All CML *Requesting Agents* (RAs) in CMN-700 require a *Requesting Agent System Address Map* (RA SAM) to determine the target *Home Agent ID* (HAID). This HAID is used as the target ID to route the Request.

The RA SAM uses configuration registers to specify address regions and corresponding HAIDs. Each address region is configured by programming the base address and corresponding size of the address region (or programming the address limit). Each valid address region is marked using a valid bit. The incoming address is compared against programmed valid address regions to generate a specific HAID.

The following figure shows a RA SAM block diagram.

Figure 3-41: RA SAM block diagram



Address region requirements

Each of the programmed address region sizes must be a power of two and must be naturally-aligned to its size. For example, a 1GB partition must start at 1GB boundary. That is, 0GB-1GB or 1GB-2GB and so forth, but it cannot start from 1.5GB or 2.5GB.

For details on how protocol messages are routed based on the RA SAM programming, see 3.4.13 Cross chip routing and ID mapping on page 184.

3.4.6 RN SAM

Transactions from an RN must pass through an RN SAM to generate a CHI target ID. The target ID is used to send the flit to the correct target node in the mesh.

CMN-700 RN-Ds, RN-Fs, RN-Is, and CCG HAs use an RN SAM that is internal to the interconnect.

When multiple RN-Fs are connected using CAL, only on RN SAM exists per CAL.

The RN SAM uses two characteristics of a transaction to map requests to downstream target nodes:

- The *Physical Address* (PA) of the request
- Whether the request is a DVM operation, a PrefetchTgt operation, or neither

The RN SAM also has a defined default target, the HN-D. It uses the default target if the preceding characteristics do not result in a match or the RN SAM has not been programmed yet.


Note

In your final RN SAM configuration, the boot code region must not change from the HN-D.

Software can configure the mapping structure for addressable requests. To configure the RN SAM, you define discrete regions of your address map and program them in the RN SAM registers. You also specify the target or group of targets for transactions to all addresses in that region.

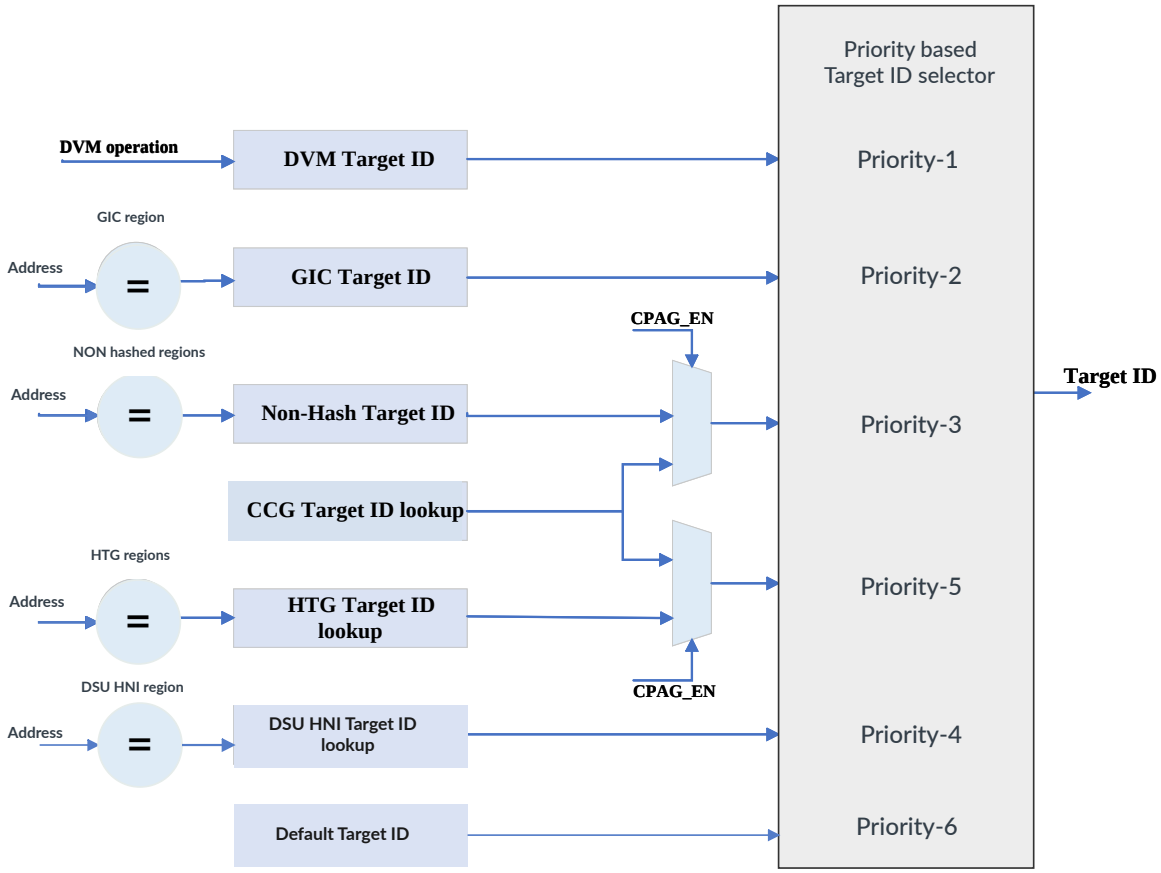
The RN SAM programming sequence is described in 4.4.3 RN SAM and HN-F SAM programming on page 1121.

For programming examples of different memory map configurations in the RNSAM and HNSAM, see 4.4.3.2 SAM programming examples on page 1123 .

3.4.6.1 Target IDs

The following describes how the target IDs is determined.

Figure 3-42: RN SAM target ID selection policy



RN SAM target ID selection policy

RN SAMs support priority-based target ID selection. The order of priority is when selecting a target ID is:

1. *Generic Interrupt Controller* (GIC) memory region (highest priority)
2. Non-hashed memory region
3. Hashed memory region or non-hashed mode of hashed memory region. These options are mutually exclusive
4. Default memory region (lowest priority)

DVM target ID

DVM transactions are assigned the DVM target ID. RN SAMs define the nodeID of the corresponding Distributed DVM domain target HN-D/HN-V/HN-T as the DVM target ID. DVM target ID is programmable at boot time which overrides the default DVM target ID.

GIC target ID

RN SAMs support a GIC memory region which can be used to select GIC-related addresses to a specific target ID. The GIC region can overlap with hashed and non-hashed regions.

Non-hashed regions

A given memory partition is assigned to an individual targetID (non-hashed). Up to 64 non-hashed regions are supported. The I/O space (HN-I, HN-D, HN-P, HN-V, and HN-T) is intended to be the target of non-hashed regions.

Hashed regions

A given memory partition is distributed (hashed) across many target devices. Hashed Target Regions (HTGs) support up to 32 hashed regions (RNSAM_NUM_HTG <= 32). HTG regions support the following configurations:

- System Cache Groups (SCG): HTG is programmed to support DRAM space to hash across HN-F based on the address
- HN-P and CCG hashing for PCIe traffic: HTG is programmed to support PCIe traffic to hash across HN-P and CCG nodes based on the AXID.

Hashed and non-hashed configuration scenarios

- A hashed region can overlap with a non-hashed region. Whether a given region is configured as hashed or non-hashed affects the target ID selection policy for that address range.
- Using an HTG region register for a single HN-I, HN-D, HN-P, and HN-T target. This scenario might be useful if all the non-hashed region registers have already been used. The region can optionally be classified as a non-hashed region (except for HTG region 0).
- Using a non-hashed region register for an HN-F target. This scenario might be useful if all the HTG region registers have already been used. For target ID selection purposes, this HN-F is classified as a non-hashed region

Default target ID

Until the RN SAM has been fully programmed, all addressable transactions use the default target ID. RN SAMs define the HN-D node ID as their default target ID. After the RN SAM has been programmed, the RN SAM only uses the default target ID for addressable requests that do not fall within one of the programmed address regions.

3.4.6.2 Memory region requirements

The following describes the RN SAM memory region requirements. SAM supports two mutually exclusive mechanisms to define a memory region. The `RNSAM_HTG_RCOMP_EN` parameter determines the mechanism that is used for hashed regions.

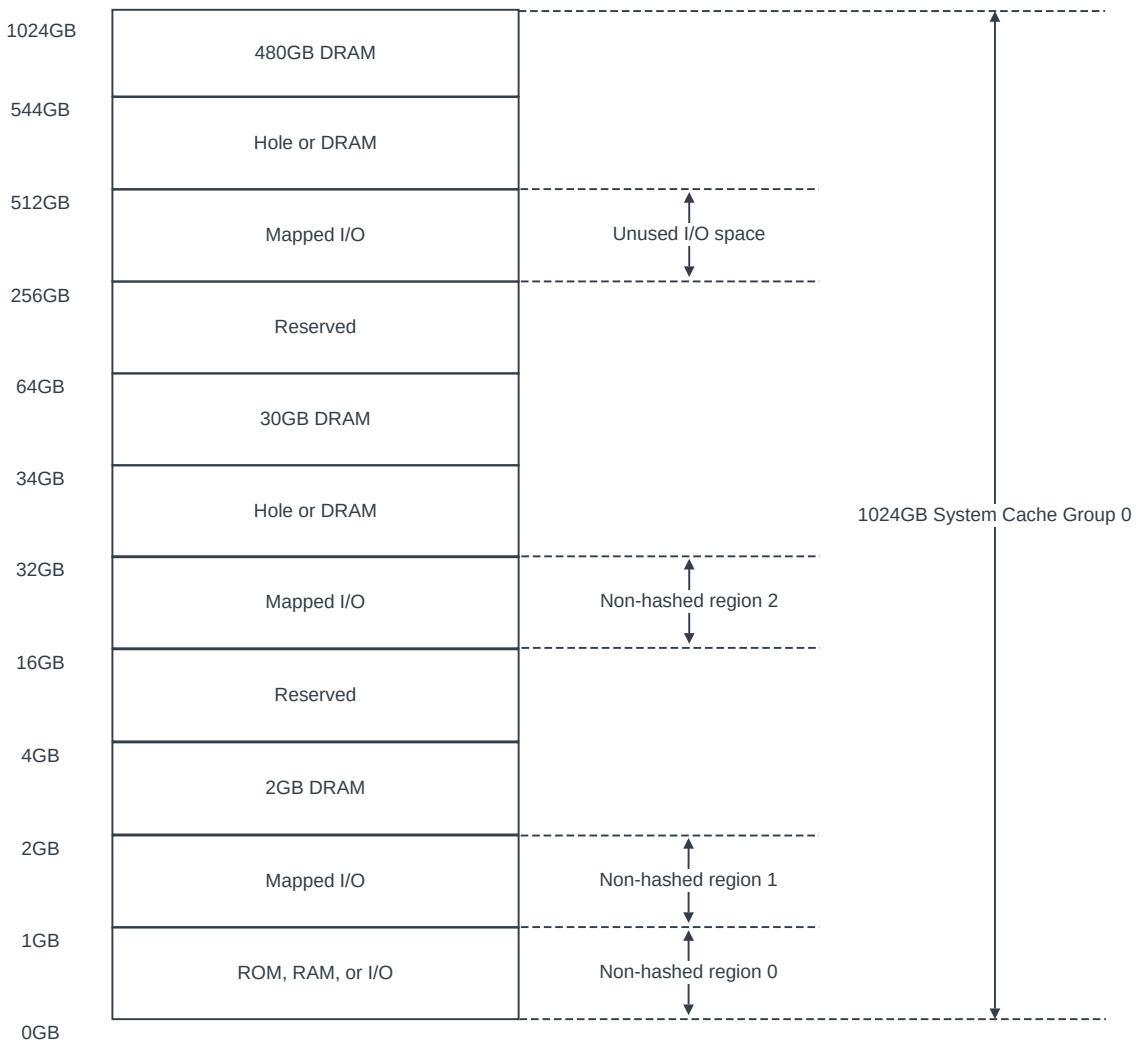
- Configurable base address & region size (`RNSAM_HTG_RCOMP_EN` = 0) - Legacy CMN mode:
  - Each of the programmed region sizes must be a power of two and the partition must be size aligned. The region size can range from 64KB–4PB. For example, a 1GB partition must start at a 1GB-aligned boundary.

    - Legacy CMN mode is supported for all the memory regions
    - Minimum size GIC region is 64KB
    - Minimum size for other regions, hashed, non-hashed, is 64MB

- Configurable lower address & upper address (`RNSAM_HTG_RCOMP_EN` = 1):
    - No restrictions on size of the region
    - Range comparison mode is supported for all memory regions except for GIC region, which always uses an aligned base address and region size.
    - Minimum size for the region is defined using user parameter (`RNSAM_HTG_RCOMP_LSB` = [16-26])
        - `RNSAM_HTG_RCOMP_LSB` = 16, defines minimum memory size = 64KB
        - `RNSAM_HTG_RCOMP_LSB` = 17, defines minimum memory size = 128KB
        - `RNSAM_HTG_RCOMP_LSB` = 20, defines minimum memory size = 1MB
        - `RNSAM_HTG_RCOMP_LSB` = 26, defines minimum memory size = 64MB
    - Non-hashed regions have separate parameters to define memory regions:
        - `RNSAM_NONHASH_RCOMP_EN`
        - `RNSAM_NONHASH_RCOMP_LSB`

Legacy CMN RN SAM mode still supports complex memory maps where DRAM region sizes are not a power of two or are not size aligned. For example, the following figure shows a memory map where the entire address is assigned to a hashed region. Then, non-hashed regions can be individually programmed, because of their higher target ID selection priority. Software must prevent accesses to addresses in a hashed region that physical memory does not back. See the *Principles of Arm® Memory Maps White Paper*.

Figure 3-43: Example memory map



For more information about RN SAM configuration register memory partition sizes, see 3.4.9 SAM memory region size configuration on page 173.

3.4.6.3 Hashed Target Groups

Hashed Target Groups (HTGs) are configured to support:

- Hashing across HN-F nodes using *System Cache Groups* (SCGs) and address-based hashing
- Hashing across HN-P and CCG nodes for PCIe traffic and AXID based hashing

SCGs

An SCG is a group of HN-Fs that share a contiguous address region. However, the addresses that are covered by each HN-F in an SCG are mutually exclusive. An HN-F belonging to an SCG is selected as the target based on a hash function.

The lowest four HTGs, HTGs 0-3, are only used for SCGs. This restriction preserves backward compatibility with legacy CMN products. The remaining HTGs can be used either for SCGs, or for HN-P or CCG PCIe hashing. SCGs support a maximum of 128 HN-F target IDs (or 256 total HN-Fs with CAL2). You can configure the HN-F target IDs into one SCG or distribute them across multiple SCGs.

SCGs support different hashing mechanisms. For all hashing mechanisms other than the default mechanism, you must enable the required hashing mechanism using its associated configuration parameter. The hashing mechanism that each SCG uses is selected from the mechanisms that are enabled in the implementation by RN SAM programming. CMN-700 supports the following hashing mechanisms:

- Power of two hashing. This mechanism is the default.
- Non-power of two hashing
- Hierarchical hashing
- User-defined hashing

SCG: Power of two hashing, legacy CMN mode

Power of two hashing supports hashing over 1, 2, 4, 8, 16, 32, 64, 128, or 256 HN-Fs. The hash function uses bits [MSB:6] of the PA of the request. To use power of two hashing over 256 HN-F nodes, you must enable CAL mode. For more information, see 3.4.6.6 HN-F with CAL support on page 132.

If CMN-700 has been configured to implement fewer than 52 PA bits, the unused upper bits are assumed to be zero. The hash algorithm calculates a pointer in the HN-F ID table in the RN SAM. The hash function is explicitly given in the following list. All numbers on the right-hand side of the equations in the list are bit positions within the PA. For example, 17 corresponds to PA bit[17]. In the equations, ^ represents XOR.

- Two HN-Fs:
  - Number of bits in select: 1
  - select [0] = (6^7^8^...^51)
- Four HN-Fs:
  - Number of bits in select: 2
  - select [0] = (6^8^10^...^50)
  - select [1] = (7^9^11^...^51)
- Eight HN-Fs:
  - Number of bits in select: 3
  - select [0] = (6^9^12^...^51)

    - select [1] = (7^10^13^...^49)
    - select [2] = (8^11^14^...^50)
- 16 HN-Fs:
    - Number of bits in select: 4
    - select [0] = (6^10^14^...^50)
    - select [1] = (7^11^15^...^51)
    - select [2] = (8^12^16^...^48)
    - select [3] = (9^13^17^...^49)
- 32 HN-Fs:
    - Number of bits in select: 5
    - select [0] = (6^11^16^...^51)
    - select [1] = (7^12^17^...^47)
    - select [2] = (8^13^18^...^48)
    - select [3] = (9^14^19^...^49)
    - select [4] = (10^15^20^...^50)
- 64 HN-Fs:
    - Number of bits in select: 6
    - select [0] = (6^12^18^...^48)
    - select [1] = (7^13^19^...^49)
    - select [2] = (8^14^20^...^50)
    - select [3] = (9^15^21^...^51)
    - select [4] = (10^16^22^...^46)
    - select [5] = (11^17^23^...^47)
- 128 HN-Fs
    - Number of bits in select: 7
    - select [0] = (6^13^20^...^48)
    - select [1] = (7^14^21^... ^49)
    - select [2] = (8^15^22^...^50)
    - select [3] = (9^16^23^...^51)
    - select [4] = (10^17^24^...^45)
    - select [5] = (11^18^25^...^46)
    - select [6] = (12^19^26^...^47)
- 256 HN-Fs:
    - Number of bits in select: 8
    - select [0] = (6^14^22^...^46)

- select [1] = (7^15^23^...^47)
- select [2] = (8^16^24^...^48)
- select [3] = (9^17^25^...^49)
- select [4] = (10^18^26^...^50)
- select [5] = (11^19^27^...^51)
- select [6] = (12^20^28^...^44)
- select [7] = (13^21^29^...^45)

SCG: Non-power of two hashing

If you enable CAL mode, non-power of two hashing supports hashing over 2-254 HN-F nodes. If CAL mode is disabled, it is only possible to hash over 2-127 HN-F nodes. For more information about CAL mode, see 3.4.6.6 HN-F with CAL support on page 132.

In this mode, TargetID index is computed through 12-bit hash followed by modulo operation. In this hashing mode, DRAM memory distribution across HN-Fs is non-uniform. To enable this hashing type, use the `RNSAM_NP2_EN` parameter. When using non-power of two-hashing, direct SN mapping per HN-F is not supported.

SCG: Hierarchical hashing

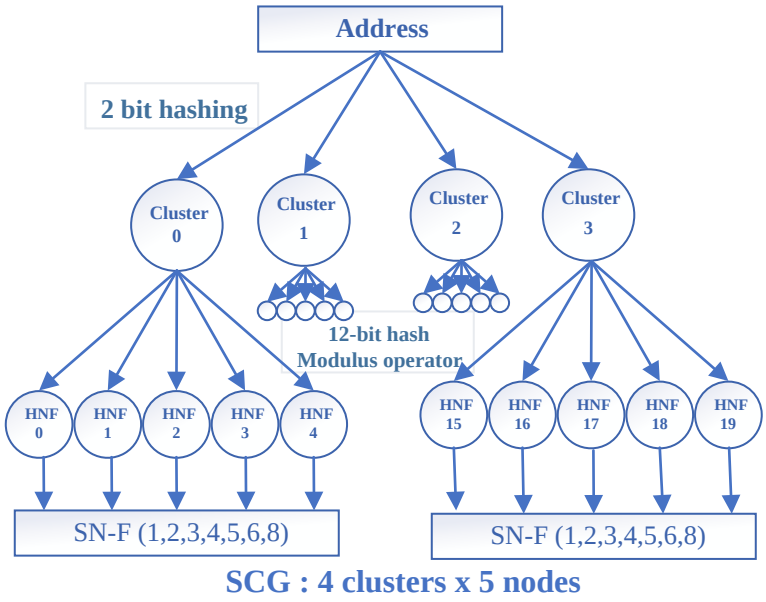
If you use hierarchical hashing, HN-Fs are grouped hierarchically in two levels. In the first level of hierarchy, HN-Fs are grouped into clusters and the addresses are hashed across the clusters. In the second level of hierarchy, addresses are hashed across the HN-F nodes in a cluster. To enable hierarchical hashing, use the `RNSAM_HIER_HASH_EN` parameter.

The first level of hierarchy supports 1, 2, 4, 8, 16, or 32 clusters. At the second level of hierarchy, each cluster supports a maximum of 32 HN-Fs and non-power of two numbers of HN-Fs are allowed.

You can configure the number of clusters and number of HN-F nodes per cluster at boot time for each SCG. In a single SCG, each cluster contains the same number of HN-F nodes.

In hierarchical hashing, the physical memory address that is hashed at the first level of hierarchy is sent to a single cluster within the SCG. Memory controllers downstream of HN-Fs must be configured to support either direct mapping per cluster or direct mapping per SCG. When using hierarchical hashing, direct SN mapping per HN-F is not supported. Also, the hierarchical hashing scheme can cause unused sets in the HN-F SLC and SF in certain use cases. For more information about mitigating unused sets, see 3.4.7.4 HN-F SLC and SF flexibile addressing on page 160.

Figure 3-44: Hierarchical hashing configuration



Non-power of two hash function

The non-power of two hash function is used for both Non-power of two hashing and for the second level of hierarchical hashing.

First, the input physical address, Addr[51:6], has lower bits shifted out according to the hashed_target_grp_hash_cntl_regN.htg_regionN_hier_enable_address_striping setting for the target SCG 'N'. For SCGs using Non-power of two hashing, this value can be set to 3'b0 to use the full Addr[51:6].

For SCGs using Hierarchical hashing, this value should be set to log2(number_of_clusters) to shift out the LSBs Addr[n:6] to achieve better HN-F and SLC set-ID distribution. For physical address spaces less than 52 bits, the upper bits are treated as zeros. The resulting address can be referred to as Addr'[51:6].

Next, the select[6:0] bits, used as a pointer into the SCG's HN-F ID table, are determined as follows:

- hash12[11:0] = {2'b00,Addr'[51:42]} ^ Addr'[41:30] ^ Addr'[29:18] ^ Addr'[17:6]
- select[6:0] = ({hash12[11] ^ hash12[0],hash12[10] ^ hash12[1],hash12[9] ^ hash12[2],hash12[8] ^ hash12[3],hash12[7] ^ hash12[4],hash12[6] ^ hash12[5],hash12[5:0]} x num_hnf) >> 12

Where num_hnf is either:

- For Non-power of two hashing
  - The number of HN-F nodes in the SCG. Each HN-F attached to a CAL instance is counted, including when CAL mode is enabled for the SCG.
- For Hierarchical hashing

- The number of HN-F nodes per cluster. Each HN-F attached to a CAL instance is counted, including when CAL mode is enabled for the SCG.

For SCGs using non-power of two hashing, the select[6:0] result is added to the SCG's base pointer into the HN-F target ID table to determine the table index of the target HN-F node.

For SCGs using hierarchical hashing, the select[6:0] result is combined with the first-level hashing result. The first-level result comes from power of two hashing applied to the request address to choose the target cluster. The final offset result is added to the SCG's base pointer into the HN-F target ID table to determine the table index of the target HN-F node. The final offset result is calculated as follows:

offset = (first-level_result * hnfs_per_cluster) + select[6:0]

When CAL mode is enabled for an SCG, the select[6:0] result is used differently to determine the table index of the target HN-F node, and to modify the table entry to generate the final HN-F target ID value. For more information about CAL mode, see 3.4.6.6 HN-F with CAL support on page 132

SCG: User-defined hashing logic

The RTL hash modules can be selectively replaced to implement user-defined hash logic. RN SAM implements user-defined configuration registers to be used for the user-defined hash logic. To set the number of enabled user-defined registers, use the `RNSAM_CUSTOM_REGS` parameter. For more information about modifying the RTL hash modules, see the *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*.

Secondary memory regions for HTGs

Each HTG supports two memory regions. If an incoming address matches either of the two programmed, valid regions, the RN SAM selects the corresponding target ID of the HTG from the target ID table. The secondary range definition mode that is used follows the value of the `RNSAM_HTG_RCOMP_EN` parameter, similar to the primary region definition. If the primary region for an SCG is set to be in non-hashed mode, the secondary region is also set to be in non-hashed mode.

SCG: target ID table configuration

Each SCG is associated with a set of HN-F nodes, and their node IDs are defined in a hashed target ID table. Alternatively, SCGs, except for SCG0, can be configured to use non-hashed mode. In non-hashed mode, the SCG can contain a single target HN, and uses separate registers from the hashed target ID table to identify the node ID.

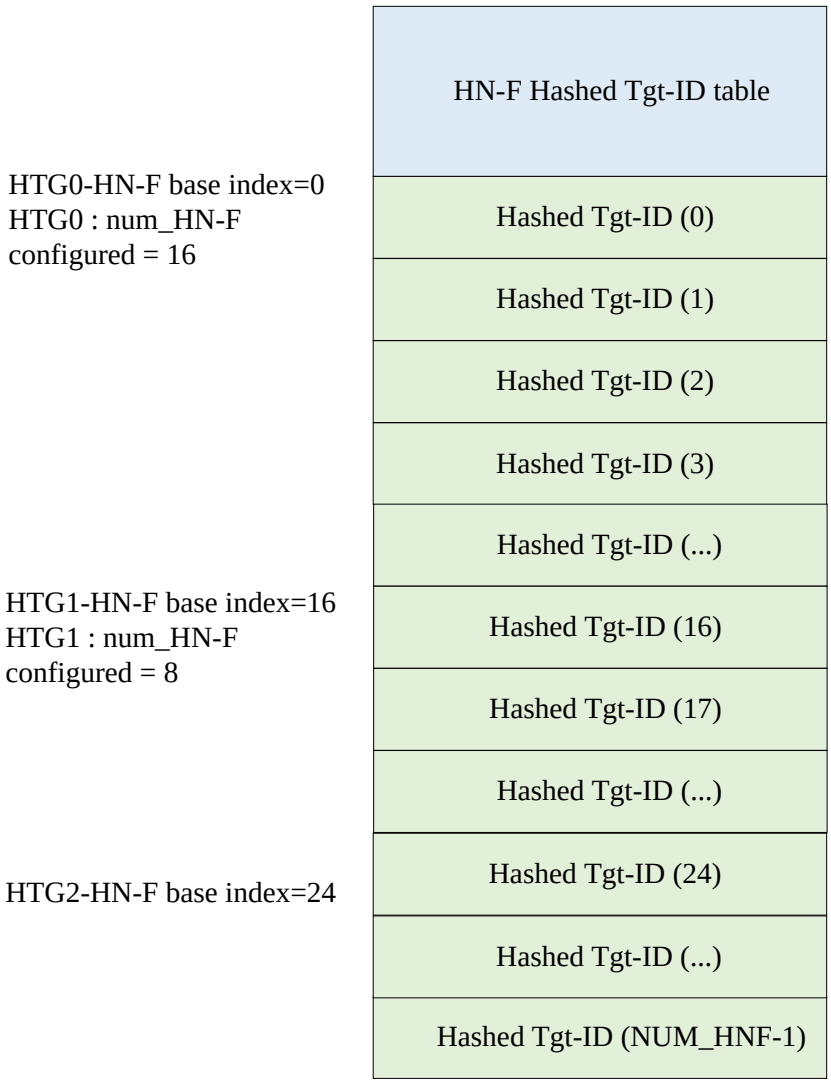
Each SCG with multiple HN-F targets consumes hashed target ID table entries from a base index for that SCG, up to the number of HN-F nodes that are assigned to that SCG. The base index for an SCG, is specified using the `RNSAM_FLEX_TGTID_EN` parameter, which is enabled by default.

- Flexible target ID base indexes (`RNSAM_FLEX_TGTID_EN` = 1). At reset, all the SCG table base indexes point to zero. The table base index of each SCG is derived in a linked list approach, based on the SCG programming:
  SCG[n] base index = (SCG[n-1] base index) + (SCG [n-1] number of HN-Fs)

For the following configuration types, this mechanism must be used and the RNSAM_FLEX_TGTID_EN parameter must be set:

- Configurations with more than four SCGs or HTGs (RNSAM_NUM_HTG parameter > 4)
- Configurations with more than 64 HN-F nodes

Figure 3-45: Example of HN-F flexible target ID configuration



- Hardcoded base index for each SCG (legacy CMN mode, RNSAM_FLEX_TGTID_EN = 0). The legacy CMN mode only applies to configurations where RNSAM_NUM_HTG = SCG = 4, and the number of HN-F nodes is less than or equal to 64. The following section and tables show how HN-F node IDs can be populated into the hashed target ID table.

When RNSAM_FLEX_TGTID_EN = 0, the following table shows the restrictions on SCG selection when there are 16, 32, or 64 HN-Fs in a given SCG.

Table 3-23: Permitted allocation of HN-Fs into SCGs (RNSAM_FLEX_TGTID_EN = 0)

SysCacheGroup HN-F or SN-F counts	1 HN-F	2 HN-F	4 HN-F	8 HN-F	16 HN-F	32 HN-F	64 HN-F
SysCacheGroup 0	Y	Y	Y	Y	Y	Y	Y
SysCacheGroup 2	Y	Y	Y	Y	Y	Y	N
SysCacheGroup 1	Y	Y	Y	Y	Y	N	N
SysCacheGroup 3	Y	Y	Y	Y	Y	N	N

The RN SAM supports up to 64 hashed HN-F and SN-F target IDs without using CAL mode when RNSAM_FLEX_TGTID_EN = 0. In this case, you can include up to 64 unique hashed target IDs in the RN SAM SCG target nodeID registers.

The nodeIDs in sys_cache_grp_[hn, sn]_nodeid_reg<X> registers are shared between all the SCGs. Therefore, the number of nodeIDs that are available for each SCG depends on the number of HN-Fs or CALs. The following algorithm determines the distribution of nodeIDs.

SCG0 NodeID0 to nodeID[n - 1]
SCG1 NodeID[n / 4] to nodeID[(n / 2) - 1]
SCG2 NodeID[n / 2] to nodeID[n - 1]
SCG3 NodeID[n x 3 / 4] to nodeID[n - 1]

In the preceding algorithm, n represents the total number of hashed target IDs in the SAM.

If SCG0 uses all the available nodeIDs, then SCG1, SCG2, and SCG3 must not be used. If SCG0 only uses nodeID0 through nodeID[(n / 2) - 1], then SCG1 cannot be used. However, in this case, you can use SCG2 and SCG3 with (n / 4) nodeIDs in each of the SCGs.

For example, the following table shows the register and nodeID allocation for each SCG in a system with 64 hashed target IDs and the RNSAM_FLEX_TGTID_EN parameter set to 0.

Table 3-24: RN SAM SCG target ID programming for 64 hashed targets (RNSAM_FLEX_TGTID_EN = 0)

SCG target ID registers (64 hashed targets)	Number of HN-Fs per SCG target ID table				
	64	32	16	8	4,2,1
SCG CAL mode supported	No	Yes	Yes	Yes	Yes
sys_cache_grp_hn_nodeid_reg0	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs
sys_cache_grp_hn_nodeid_reg1					-
sys_cache_grp_hn_nodeid_reg2				-	-
sys_cache_grp_hn_nodeid_reg3				-	-
sys_cache_grp_hn_nodeid_reg4			SCG1_NIDs	SCG1_NIDs	SCG1_NIDs
sys_cache_grp_hn_nodeid_reg5					-
sys_cache_grp_hn_nodeid_reg6				-	-
sys_cache_grp_hn_nodeid_reg7				-	-
sys_cache_grp_hn_nodeid_reg8		SCG2_NIDs	SCG2_NIDs	SCG2_NIDs	SCG2_NIDs

SCG target ID registers (64 hashed targets)	Number of HN-Fs per SCG target ID table				
	64	32	16	8	4,2,1
sys_cache_grp_hn_nodeid_reg9					-
sys_cache_grp_hn_nodeid_reg10				-	-
sys_cache_grp_hn_nodeid_reg11				-	-
sys_cache_grp_hn_nodeid_reg12			SCG3_NIDs	SCG3_NIDs	SCG3_NIDs
sys_cache_grp_hn_nodeid_reg13					-
sys_cache_grp_hn_nodeid_reg14				-	-
sys_cache_grp_hn_nodeid_reg15				-	-

The following table shows the register and nodeID allocation for each SCG in a system with 16 hashed target IDs and the `RNSAM_FLEX_TGTID_EN` parameter set to 0.

Table 3-25: RN SAM SCG target ID programming for 16 hashed targets (`RNSAM_FLEX_TGTID_EN` = 0)

SCG target ID registers (16 hashed targets)	Number of HN-Fs per SCG target ID table			
	16	8	4	2, 1
sys_cache_grp_hn_nodeid_reg0	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs
				-
sys_cache_grp_hn_nodeid_reg1			SCG1_NIDs	SCG1_NIDs
				-
sys_cache_grp_hn_nodeid_reg2		SCG2_NIDs	SCG2_NIDs	SCG2_NIDs
				-
sys_cache_grp_hn_nodeid_reg3			SCG3_NIDs	SCG3_NIDs

The following table shows the register and nodeID allocation for each SCG in a system with eight hashed target IDs and the `RNSAM_FLEX_TGTID_EN` parameter set to 0.

Table 3-26: RN SAM SCG target ID programming for eight hashed targets (`RNSAM_FLEX_TGTID_EN` = 0)

SCG target ID registers (8 hashed targets)	Number of HN-Fs per SCG target ID table			
	8	4	2	1
sys_cache_grp_hn_nodeid_reg0	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs
				-
			SCG1_NIDs	SCG1_NIDs
				-
sys_cache_grp_hn_nodeid_reg1		SCG2_NIDs	SCG2_NIDs	SCG2_NIDs
				-
			SCG3_NIDs	SCG3_NIDs
				-

The following table shows the register and nodeID allocation for each SCG in a system with four hashed target IDs and the `RNSAM_FLEX_TGTID_EN` parameter set to 0.

Table 3-27: RN SAM SCG target ID programming for four hashed targets (`RNSAM_FLEX_TGTID_EN` = 0)

SCG target ID registers (4 hashed targets)	Number of HN-Fs per SCG target ID table		
	4	2	1
sys_cache_grp_hn_nodeid_reg0	SCG0_NIDs	SCG0_NIDs	SCG0_NIDs
			SCG1_NIDs
		SCG2_NIDs	SCG2_NIDs
			SCG3_NIDs

The hashed target ID allocation in the preceding tables is also applicable to SN target IDs.

The following table shows an example mapping of HN-Fs to SCGs when the `RNSAM_FLEX_TGTID_EN` parameter is set to 0.

Table 3-28: 25 HN-Fs to three SCGs programming example (`RNSAM_FLEX_TGTID_EN` = 0)

SCG number	Number of HN-Fs	Node ID
SCG0	16	NID0-15
SCG2	8	NID16-23
SCG3	1	NID24

The following table shows example programming for 25 HN-Fs when the `RNSAM_FLEX_TGTID_EN` parameter is set to 0.

Table 3-29: Example programming for 25 HN-Fs (`RNSAM_FLEX_TGTID_EN` = 0)

SCG target ID registers	Number of HN-Fs per SCG			
	32	16	8	1
sys_cache_grp_hn_nodeid_reg0	-	SCG0 NID0-15	-	-
sys_cache_grp_hn_nodeid_reg1				
sys_cache_grp_hn_nodeid_reg2				
sys_cache_grp_hn_nodeid_reg3				
sys_cache_grp_hn_nodeid_reg4		-	SCG2 NID16-23	
sys_cache_grp_hn_nodeid_reg5				
sys_cache_grp_hn_nodeid_reg6			-	SCG3 NID24
sys_cache_grp_hn_nodeid_reg7				-

Hashing across HN-P and CCG nodes (PCIe traffic, AxID-based hashing)

In this mode, HTGs hash incoming PCIe traffic from RN-I or RN-D nodes across multiple HN-P or CCG nodes. The hashing is based on two factors:

- The AxID, in other words the ARID or AWID of the request that is presented to the RN-I or RN-D ACE-Lite interface
- Which of the three ACE-Lite interfaces on the RN-I or RN-D node received the request

HN-P and CCG target IDs that are associated with each HTG are derived through flexible base indexes similar to HN-F target IDs. However, there is a separate target ID table.

If you use AxID-based hashing, then legacy CMN mode, using hardcoded base indexes, is not supported. To enable hashing across HN-P and CCG nodes, set the following parameter values:

- `RNSAM_AXID_HASH_EN` = 1
- `RNSAM_FLEX_TGTID_EN` = 1

The hashing input is defined as:

rnsam_axid_in_port = {port_id, AXID} = {2 bits, 32 bits} = 34 bits

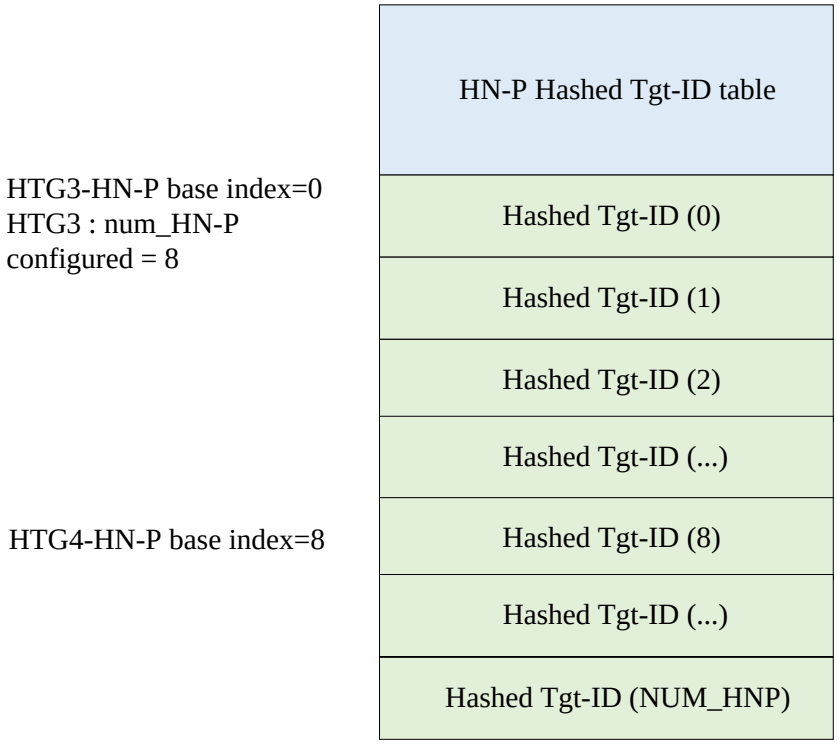
The AxID interface value is zero-extended when the AxID interface width is less than 32 bits.

An HTG supports hashing over 1, 2, 4, 8, 16, 32 HN-Ps/CCGs, using the full AXID of the request and AXI port identifier. The hash algorithm calculates a pointer in the HN-P ID table in the RN SAM. The hash function is explicitly given in the following list:

- Two HN-Ps or CCGs:
  - Number of bits in select: 1
  - select [0] = (0^1^2^...^33)
- Four HN-Ps or CCGs:
  - Number of bits in select: 2
  - select [0] = (0^2^4^...^32)
  - select [1] = (1^3^5^...^33)
- Eight HN-Ps or CCGs:
  - Number of bits in select: 3
  - select [0] = (0^3^6^...^33)
  - select [1] = (1^4^7^...^31)
  - select [2] = (2^5^8^...^32)
- 16 HN-Ps or CCGs:
  - Number of bits in select: 4
  - select [0] = (0^4^8^...^32)
  - select [1] = (1^5^9^...^33)
  - select [2] = (2^6^10^...^30)
  - select [3] = (3^7^11^...^31)
- 32 HN-Ps or CCGs:
  - Number of bits in select: 5
  - select [0] = (0^5^10^...^30)
  - select [1] = (1^6^11^...^31)
  - select [2] = (2^7^12^...^32)
  - select [3] = (3^8^13^...^33)

- select [4] = (4^9^14^...^29)

Figure 3-46: Example of HN-P flexible target ID configuration



3.4.6.4 Address range-based QoS override and PrefetchTarget support

CMN-700 provides resources that can override the QoS value of requests and facilitate PrefetchTarget operations to specific address ranges.

The value of the `RNSAM_NUM_QOS_REGIONS` configuration parameter and the value of a bit within the sam_qos_mem_region_reg* registers determine which features are enabled. The following table shows the possible modes and their respective values.

Table 3-30: QoS override and PrefetchTarget mode configuration values

`RNSAM_NUM_QOS_REGIONS` value	sn_tgtid_override value	
	0	1
0-7	QoS override mode only enabled	
8	QoS override mode only enabled	QoS override mode and PrefetchTarget mode enabled

The QoS override mode and the PrefetchTarget mode are described in the following sections.

Address range-based QoS override

You can program the RN SAM to override the QoS value of requests from RNs to HNs that target certain memory regions. This feature is present in all instances of the RN SAM inside MXP, RN-I, RN-D, and CML-HA.

You can use this feature to change the priority of traffic targeting high-priority or low-priority memory or I/O devices. With this feature, requests that pass through the same QoS regulators can have different QoS values.

You configure the number of override memory regions using the `RNSAM_NUM_QOS_REGIONS` parameter. Each region corresponds to one of the sam_qos_mem_region_reg* registers. Each of these registers contains a bit to indicate the following details for the region:

- Region valid: indicates that the programmed memory region is valid for comparison
- Region base address
- Region size
- Region QoS value
- Override bit: determines whether override occurs for that memory region or not

The QoS regions follow the range definition method determined by the `HNSAM_RCOMP_EN` parameter.

`HNSAM_RCOMP_EN` = 0
: The regions have the same base address and size properties as hashed and non-hashed regions

`HNSAM_RCOMP_EN` = 1
: There is an additional register per region to define the region's end address

When you enable the Address range-based QoS override feature, CMN-700 compares the incoming address against the valid QoS memory regions. If the address matches any of the programmed valid addresses, the request flit uses the corresponding QoS override value.

The memory regions that you specify for override are independent of the hashed and non-hashed memory regions in the RN SAM. Therefore, QoS override regions can overlap with either of the hashed or non-hashed regions. However, two QoS override regions must not overlap with each other.

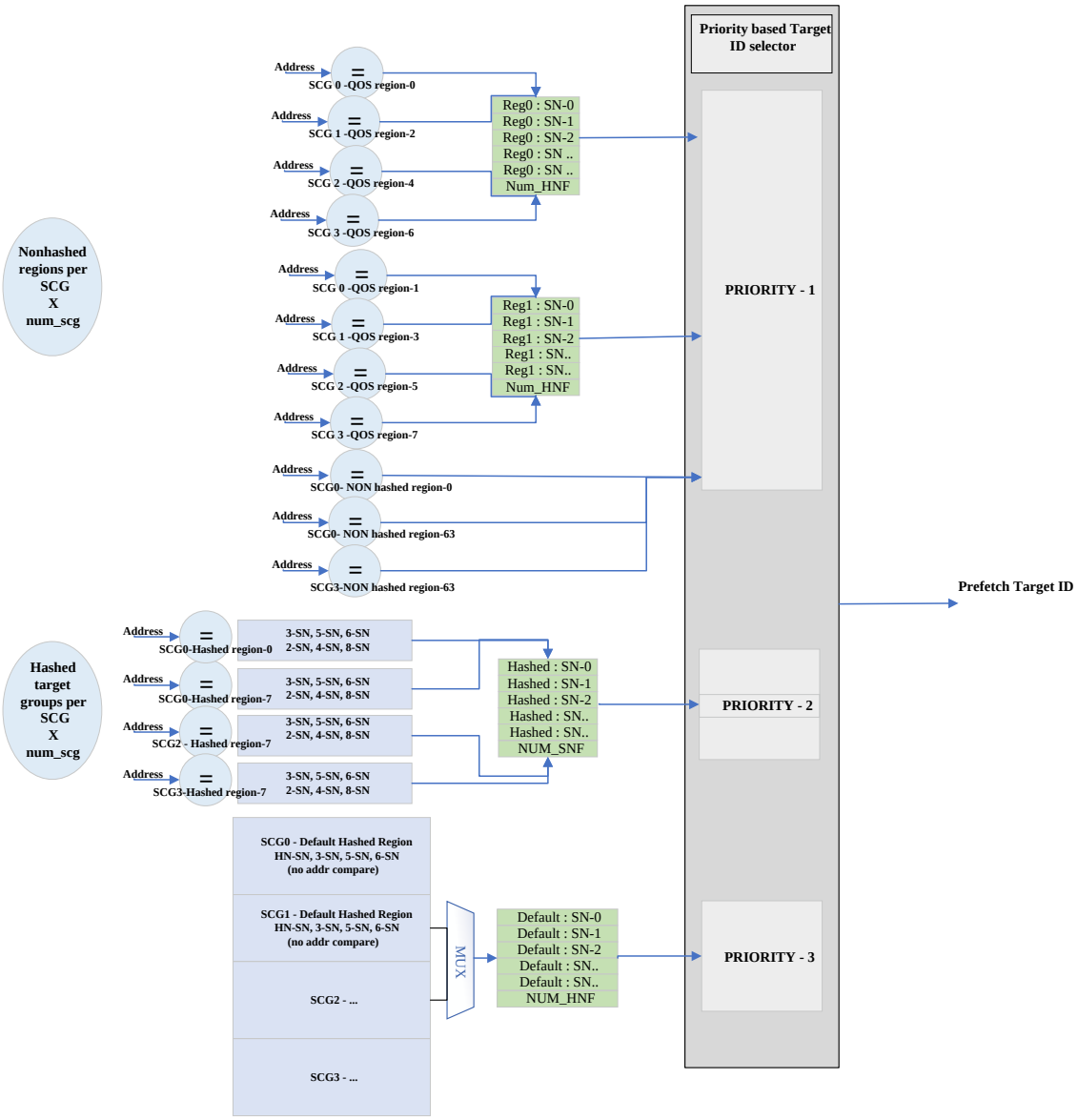
The RN SAM QoS memory regions and QoS override do not apply to the GIC memory region.Therefore, you must not specify the GIC memory region as a target for QoS override.

Additionally, request QoS override functionality is available using the RN SAM PrefetchTarget Non-hashed regions, as 3.4.6.5 Support for PrefetchTarget operations in RN SAM on page 128 describes.

3.4.6.5 Support for PrefetchTarget operations in RN SAM

The RN SAM supports CHI PrefetchTarget operations. These operations are sent from RN-F directly to SN-F, bypassing the HN-F. The RN SAM prefetchTarget target ID generation must comprehend both the RN SAM HN-F target ID generation and HN-F SAM SN target ID generation.

Figure 3-47: RNSAM Prefetch target ID



To support PrefetchTarget operations, the RN SAM integrates the functionality of HN-F SAM to determine the appropriate SN-F target ID for a given address. RN SAMs that integrate PrefetchTarget support only use the SN-F target ID for PrefetchTarget requests. The PrefetchTarget RN SAM programming must match the HN-F SAM for SN-F target IDs. PrefetchTarget support in the RN SAMs is enabled by the `RNSAM_PREFETCH_EN` parameter. When enabled, the RN SAM

supports PrefetchTarget IDs for a maximum of 8 SCGs, determined by the `RNSAM_PFTGT_NUM_SCG` parameter.

PrefetchTarget support is configured to match the HN-F SAM configuration for the associated address ranges:

- Direct-mapped or address-striped in the HN-F SAM (Default Hashed Region)
- Non-hashed target regions in the HN-F
- HTG region in the HN-F

PrefetchTarget operations to direct-mapped or address-striped SN-F targets

The following registers are used to program the PrefetchTarget functionality in the RN SAM:

- por_rnsam_sys_cache_grp_sn_attr
- por_rnsam_sys_cache_grp_sn_nodeid_reg{0-7}
- por_rnsam_sys_cache_grp_sn_sam_cfg{0-1}

This mode is enabled through setting the `RNSAM_PFTGT_DEF_HASHED_GRP_EN` parameter.

Support for PrefetchTarget operations to address range-based targets

The RN SAM QoS regions can be used to generate PrefetchTarget SN TargetIDs matching the cmn_hns_sam_mem_region_0-1 targets. To enable this functionality, the following parameters must be set as follows:

- `RNSAM_PFTGT_DEF_HASHED_GRP_EN` = 1
- `RNSAM_NUM_QOS_REG` = 8
- `RNSAM_PFTGT_NUM_SCG` = 4

The sn_tgtid_override bit for the matching QoS region must be set to 1. If the address of the request is within the corresponding SCG (0-3), the RN SAM generates the SN target ID as programmed in sys_cache_grp_region[0-1]_sn_nodeid_reg[0-31] registers.

The node IDs of the SNs in these registers are mapped to each SCG similarly to the node IDs of the HNs in SCGs. The following table shows the mapping of each SCG to the relevant QoS region registers.


Note

The memory regions that are programmed in the sam_qos_mem_region_regX registers must match the memory regions that are programmed in the cmn_hns_sam_memregion[0-1] registers.

Table 3-31: Mapping of SCGs to QoS region registers

SCG ID	Mapped registers
SCG0	• sam_qos_mem_region_reg0 • sam_qos_mem_region_reg1

SCG ID	Mapped registers
SCG1	• sam_qos_mem_region_reg2 • sam_qos_mem_region_reg3
SCG2	• sam_qos_mem_region_reg4 • sam_qos_mem_region_reg5
SCG3	• sam_qos_mem_region_reg6 • sam_qos_mem_region_reg7

PrefetchTarget operations based on HN-F Non-hashed regions

PrefetchTarget operations for the HN-F non-hashed memory regions are supported and assigned a unique SN-F target ID for each non-hashed region. RNSAM supports a maximum of 64 non-hashed regions per system cache group, configured via the `RNSAM_PFTGT_NUM_NONHASH_PSCG` parameter. Therefore the total number of PrefetchTarget non-hashed regions supported by the RNSAM is given by:

`RNSAM_PFTGT_NUM_SCG` X `RNSAM_PFTGT_NUM_NONHASH_PSCG`

PrefetchTarget non-hash regions also support QoS override capability. Each of the defined and valid PrefetchTarget non-hashed region address ranges can be enabled for QoS override and provide a replacement request QoS value. Any request with an address that matches a PrefetchTarget non-hashed address range can have its QoS value overridden regardless of request opcode, not just PrefetchTarget request types.

PrefetchTarget operations based on HN-F HTG regions

PrefetchTarget operations for the HN-F HTG regions are supported and assigned an SN-F through address striped hash functions. RNSAM supports a maximum of 8 hashed target regions per system cache group. The total number of PrefetchTarget hashed regions supported by the RNSAM is given by:

`RNSAM_PFTGT_NUM_SCG` X `RNSAM_PFTGT_NUM_HTG_PSCG`

Prefetch Hashed region supports the following hash functions: MOD-3/5/6 hashing, Power of two hashing (2SN, 4SN, 8SN). For more information about these hashing mechanisms see 3.4.7 HN-F SAM on page 147.

Prefetch hashed and non-hashed memory regions support two different ways of defining a memory region based on the `HNSAM_RCOMP_EN` parameter:

- Configurable base address & region size (HNSAM_RCOMP_EN = 0) – Legacy CMN mode.
  - Each of the programmed region sizes must be a power of two and the partition must be size aligned. The region size can range from 64MB–4PB. For example, a 1GB partition must start at a 1GB-aligned boundary
  - Legacy CMN mode is supported for all the memory regions.
- Configurable lower address & upper address (`HNSAM_RCOMP_EN` = 1).
  - No restrictions on size of the region
  - Minimum size for the region is defined using user parameter (`HNSAM_RCOMP_LSB` = [20-26]).

- `HNSAM_RCOMP_LSB` = 20, defines minimum memory size = 1MB
- `HNSAM_RCOMP_LSB` = 26, defines minimum memory size = 64MB

SN-TargetID lookup for the PrefetchTarget operations

PrefetchTarget operations targeting Hashed, Default hashed, and QoS regions derive the SN target ID from the configured SN Target ID tables.

HTG regions

PrefetchTarget operations targeting the HN-F HTG regions derive the SN target ID based on the address hashing mechanism, and the resultant hash index is looked up into the SN Target ID table. Each of the prefetch HTG regions has an index to SN target ID table.

Default hashed regions

Prefetch operations targeting the Default hashed regions derive the SN target ID based on the address hashing mechanism or HN-F affinity, and the resultant hash index is looked up into the Default hashed SN Target ID table. SN-F base indexes for each system cache group follow the same HN-F base index for the associated system cache group.

QoS regions

Prefetch operations targeting the QoS regions derive the SN target ID from the QoS region SN Target ID tables. QoS regions-0/2/4/6 target the Region0 SN Target ID table, and QoS regions-1/3/5/7 target the Region1 SN Target ID table. The SN target ID indexes from these tables follow the HN-F indexes of associated system cache groups.

3.4.6.6 HN-F with CAL support

CMN-700 system supports pairing two HN-Fs at an MXP port using CAL.

The HN-F NodeIDs paired at the CAL are only differentiated using the device ID (NodeID[1:0]) field in the node ID. See 3.4.1 Node ID mapping on page 101.

There are various options for assigning HN-F nodes to *System Cache Groups* (SCGs) when CALs are used.


Note

The following description of these options uses an example configuration, which is:

- Four CAL instances connect eight HN-F nodes, that is two HN-F nodes per CAL
- All eight HN-F nodes belong to the same SCG

Normal mode

In normal mode, each HN-F node ID is explicitly assigned to an SCG using the methods that 3.4.6.3 Hashed Target Groups on page 116 describes. In the example configuration, all eight HN-F node IDs are entered in the target ID registers of the SCG.
The HN-F count field for that SCG would be set to eight. This approach allows up to 128 HN-F nodes that are connected to 64 CAL instances to be assigned to an SCG.

CAL mode

In CAL mode, only one of the two HN-Fs on the CAL has its node ID entered into the SCG target ID registers. In the example configuration, only four HN-F node IDs are entered in the SCG target ID registers, one per CAL. The HN-F count field for that SCG would be set to four. This approach allows up to 256 HN-F nodes that are connected to 128 CAL instances to be assigned to an SCG.

Mixed mode

In systems with mixed CAL and non-CAL HN-Fs, a single SCG can contain a mix of CAL and non-CAL HN-Fs. The CAL HN-Fs must be individually programmed as in normal mode. Do not enable the `scg<X>_hnf_cal_mode_en` field in the `sys_cache_grp_cal_mode_reg` register or the `htg<X>_hn_cal_mode_en` field in the `hashed_target_grp_cal_mode_reg` register for this kind of SCG.

The following table lists the options.

Table 3-32: HN-F CAL/non-CAL options

Number of HN-F target IDs (configured per SCG)	CAL mode disabled (total HNs hashed)	CAL mode enabled (total HNs hashed)
2	2	4
4	4	8
8	8	16
16	16	32
32	32	64
64	64	128
128	128	256

If there are HN-F CAL instances that are configured in normal mode in your configuration, it might be necessary to modify the `RNSAM_NUM_ADD_HASHED_TGT` global RTL parameter when configuring the mesh in Socrates™. The number of HN-F CAL instances determines the number of sys_cache_grp_hn_nodeid/ hashed_target_grp_hnf_nodeid registers that are rendered in the HN-F, not the number of HN-F nodes. For the number of these registers to at least match the absolute number of HN-F nodes, you must increase the `RNSAM_NUM_ADD_HASHED_TGT` parameter value by the number of HN-F CAL instances that are present in the mesh.

The RN SAM `sys_cache_grp_cal_mode_reg` and `hashed_target_grp_cal_mode` registers contain the CAL mode enable bit for each system cache group. These registers also contain bits per-SCG to select the override bit behavior described below, as well a CAL type bit. For CMN-700, the CAL type bit must indicate the "CAL2" type.

When CAL mode is enabled, the node ID of only one of the two nodes connected to the CAL, the lesser of the two values, is entered into the applicable Target ID table. Initial node selection occurs as previously described for each hashing method. When the target node ID has been selected from the table, the DeviceID, bits [0] of the node ID, is overridden to select one of the two nodes behind the CAL.

SCG target ID selection with CAL mode: power-of-two hashing

For example, consider an SCG that is programmed to have four HN-Fs and HN-F CAL mode enabled for this region. For this SCG, the RN SAM generates eight unique target IDs according to the following function:

Number of bits in select: 3

select[0] = (6^9^12^15^18^21^24^27^30^33^36^39^42^45^48^51)
select[1] = (7^10^13^16^19^22^25^28^31^34^37^40^43^46^49)

select[2] = (8^11^14^17^20^23^26^29^32^35^38^41^44^47^50)

The select bits are used to determine the target HN-F node ID depending on the `scg*_hnf_cal_bit_override` and `htg*_hn_cal_bit_override` configuration bit values. By default, the MSB, select[2] in this example, is used to alter the DeviceID[0] of the selected HN-F TargetID value (hash_nodeID_pick[0]). While the lower bits, select[1:0] in this example, are used to pick between the programmed HN-F target IDs (four in this example).

Alternatively, the LSB, select[0] in this example, can be used to alter the DeviceID[0] of the selected HN-F TargetID value (hash_nodeID_pick[0]). While the upper bits, select[2:1] in this example, are used to pick between the programmed HN-F target IDs (four in this example).

Assuming the default "MSB" override, the HN-F target node ID is determined as follows:

Target NodeID[10:0] = {hash_nodeID_pick[10:1], hash_nodeID_pick[0] | select[2]}.

SCG target ID selection with CAL mode: non-power of two and hierarchical hashing

For SCGs using non-power of two hashing or hierarchical hashing, the select[6:0] result from the non-power of two hash function is used to determine the offset from the SCG's base index into the HN-F target ID table, and to modify the hash_nodeID_pick[10:0] retrieved from the table to generate the final HN-F target node ID.

For SCGs using non-power of two hashing with CAL mode enabled, select[6:1] is added to the SCG's base index in the HN-F target ID table. The resulting index is used to obtain hash_nodeID_pick[10:0]. The final HN-F target ID is determined as follows:

Target_NodeID[10:0] = {hash_nodeID_pick[10:1], hash_nodeID_pick[0] | select[0]}

For SCGs using hierarchical hashing with CAL mode enabled, the offset from the SCG's base index in the HN-F target ID table is determined as follows:

offset = (first-level_result * (hnfs_per_cluster / 2)) + select[6:1]

This offset is added to the SCG's base index in the HN-F target ID table, and the resulting index is used to obtain hash_nodeID_pick[10:0]. The final HN-F target ID is determined as follows:

Target_NodeID[10:0] = {hash_nodeID_pick[10:1], hash_nodeID_pick[0] | select[0]}

The following limitations apply when using CAL mode:

- This feature must only be used when the target HN-F are paired using CAL.

- RN SAM does not apply this method to SN-F Target IDs. To fully utilize CHI PrefetchTarget operations to SN-F, the paired HN-Fs must always be mapped to same SN-F or group of SN-Fs if 3-SN, 5-SN, or 6-SN hashing is used.
- Only the lowest value HN-F ID from each CAL group must be programmed in the RN SAM hashed Target ID registers.
- If an SCG contains a mix of local HN-F and CCG NodeIDs, HN-F CAL mode must not be used for that SCG.

3.4.6.7 Non-hashed region options in RN SAM

The RN SAM supports 1 - 64 unique non-hashed regions.

Each of the non-hashed regions target a single HN-F, HN-I (all flavors), or CML RA node. The address range for each non-hashed region is defined depending on the `RNSAM_NONHASH_RCOMP_EN` parameter as described previously. Matches to non-hashed regions have higher priority than matches to hashed regions.

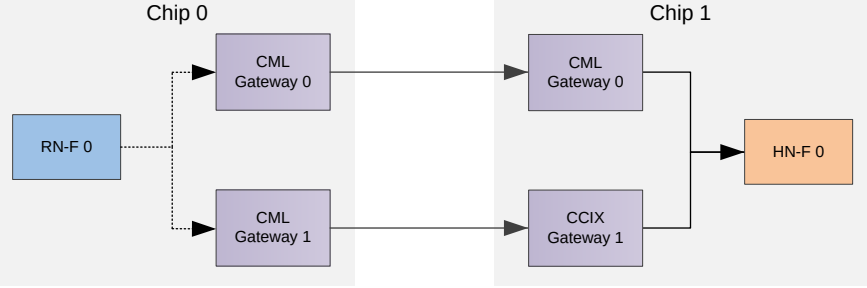
3.4.6.8 SAM support for CML Port Aggregation

RN SAM and HN-F SAM both support *CML Port Aggregation* (CPA) functionality.

Support for CPA in RN SAM

Requests from an RN to a remote chip can be striped across CCGs based on the aggregation function defined in RN SAM's *Hash Target Group* (HTG). The following figure shows an example of how requests from given RN-F can be aggregated across multiple CML gateway blocks when targeting a remote HN-F.

Figure 3-48: RN SAM CML Port Aggregation



This striping is achieved by hashing physical address bits[51:6]. The RN SAM can hash incoming addresses across up to 32 gateway blocks forming a *CML Port Aggregation Group* (CPAG). RN SAM can support up to 16 CPAGs (`RNSAM_NUM_CPA_GRP` <= 16). RN SAM also has an address mask for each CPA which can be used to remove certain bits from the hashing.

For example, to stripe the incoming address at 512B granularity, the address mask bits[51:6] can be set to `0x3FFFFFFFFF8`. With this mask setup, the logical operator AND is applied to all incoming addresses before hashing the address bits. For information about programming, see the RN SAM registers.

RN SAM supports maximum of 32 CML target IDs. Each CPAG derives the PAG target ID based on the address hashing and the hash index is looked up into the CML target ID table. Base index for each CPAG is derived based on the linked list mechanism when `RNSAM_FLEX_TGTID_EN` = 1.

RN SAM instance in RNI supports programmable CCG base index per CPAG. This enables a CCG to be part of multiple CPAGs (one CPAG for address based hashing and one CPAG for AXID based hashing).

CPAG supports following hashing ways:

- power of two (2/4/8/16/32 - way) hashing
- non-power of two (mod-3 way) hashing

Legacy CMN Mode

Legacy CMN mode for CPAG is defined when these parameters are configured to the specified values:

- `RNSAM_FLEX_TGTID_EN` = 0
- `RNSAM_NUM_CPA_GRP` = 5

In this mode, the maximum number of CML Target ID table entries is 10.

The following table shows the number of CML targets that are allowed in each CPAG when using Legacy CMN mode.

Table 3-33: Number of CML targets allowed in each CPAG (Legacy CMN mode)

CPA Group ID	Number of CCGs supported
CPAG_0	8
CPAG_1	2
CPAG_2	4
CPAG_3	2
CPAG_4	2

When using Legacy CMN mode, CPAGs use a shared target ID table with a maximum total of ten target IDs. Therefore the number of CPAGs that are available depends on the number targets that are assigned to each CPAG. For example, if CPAG 0 is hashing across eight ports, then CPAG 1, CPAG 2, and CPAG 3 must not be used. However, CPAG 4 can contain the two remaining target IDs.

The following table shows the mapping of target IDs to each CPAG when using Legacy CMN mode.

Table 3-34: Mapping of target IDs to CPAGs (Legacy CMN mode)

cml_port_aggr_grp_reg fields	CPAG_0	CPAG_1	CPAG_2	CPAG_3	CPAG_4
pag_tgtid0	Y	-	-	-	-
pag_tgtid1	Y	-	-	-	-
pag_tgtid2	Y	Y	-	-	-
pag_tgtid3	Y	Y	-	-	-
pag_tgtid4	Y	-	Y	-	-
pag_tgtid5	Y	-	Y	-	-
pag_tgtid6	Y	-	Y	Y	-
pag_tgtid7	Y	-	Y	Y	-
pag_tgtid8	-	-	-	-	Y
pag_tgtid9	-	-	-	-	Y

Support for CPA in HN-F SAM

HN-F also supports CPA for snoop requests going to a remote chip. HN-F uses the same hashing as RN SAM so that a given address always goes to the same CML gateway block for both requests and snoops.

HN-F uses the *Logical ID* (LDID) of the RN-F to determine whether CPA is enabled. As 3.4.13 Cross chip routing and ID mapping on page 184 describes, HN-F contains the LDID to physical node ID conversion table as Table 3-57: Example program on page 188 shows in the Example Programming table. This table, along with the CHI node ID and valid fields, also contains remote, cpa_en, and cpa_grpid bits. HN-F uses these bits to determine whether the RN-F is enabled to use CPA. It then sends the snoops through appropriate ports by hashing the address bits. To enable CPA for the ID of each RN-F, see 3.4.7 HN-F SAM on page 147.

AXID based CML port aggregation

CML port aggregation supports AXID based hashing for selecting the CCG port. This mode is enabled by programming axid_hash_en bit for each CPAG separately. CPAG address mask is used to mask any AXID bits AXID_MASK [31:0] = ADDR_MASK [37:6].

When enabling CPA in RN SAM and HN-F SAM, the following rules apply:

- CPA can only be used for HTG and non-hashed address ranges in the RN SAM
- CPA must not be enabled for Device memory traffic. CPA must not be enabled for an address range that can be changed from Normal to Device. This can happen with updates to page tables, where a given memory region is changed from Normal to Device.
- Each non-hashed memory range can be explicitly enabled to use CPA or use a single non-hashed target ID
- The target ID of each HTG can be explicitly enabled to use CPA
- The HN-F, when participating as a CPA target for traffic from a remote RN-F, must not receive non-CPA traffic from the same remote RN-F. This requirement means that an RN-F cannot send CPA and non-CPA traffic to the same remote HN-F.
- cml_port_aggr_grp_reg<X> registers contain the full list of CML gateway target IDs

- HTG in RN SAM supports mutiple CPA groups. Each HN-F can be programmed to associate to different CPA group.
- CPA_EN for each HN-F is specified by programming bit vector to register `sys_cache_grp_hn_cpa_en_reg` (0-63 HNFs) and `hashed_target_grp_hnf_cpa_en_reg1` (64-128 HNFs).
- Multiple CPA groups are enabled by programming each HTG `sys_cache_grp_hn_cpa_grp_reg#.enable_multi_cpa_grp_scg#` = 1'b1. CPA group per each HN-F is specified by programming group ID vector to `hashed_target_grp_cpag_perhnf_reg#`.
- If HTG is enabling only one CPA group (Legacy mode) then programming is simplified by disabling `sys_cache_grp_hn_cpa_grp_reg. enable_multi_cpa_grp_scg#` = 1'b0 and programming CPA group per each HTG `sys_cache_grp_hn_cpa_grp_reg#.cpa_grp_scg#`

For PCIe writes tunneled to the remote memory, address based hashing should be disabled. Instead HTG can be configured for AXID based hashing, where CCG NodeIDs are programmed on HN-P target ID table

AXID based hashing is not supported inside of the CPAG.

3.4.6.9 Address bit masking in the RN SAM

The CMN-700 RN SAM supports masking of address bits used for range compare and address hashing.

Address bit masking in the RN SAM can be enabled by programming the following registers:

- rnsam_hash_addr_mask_reg
- rnsam_region_cmp_addr_mask_reg

When RN SAM compares the incoming address against the programmed ranges, it uses different address bit ranges for different region types:

- For hashed and non-hashed memory regions, RN SAM uses address bits:
  - [MSB:26] when the applicable `*RCOMP_EN` parameter is 0.
  - [MSB:2^{*RCOMP_LSB}] When the applicable `*RCOMP_EN` parameter is 1
- For GIC memory region, RN SAM uses address bits [MSB:16].

By programming select bits to `0b0` in the rnsam_region_cmp_addr_mask_reg mask register, both the incoming address and the programmed address ranges can be masked off before comparison. This region mask is applied to hashed, non-hashed, and GIC memory regions.

RN SAM hashes all address bits [MSB:6] to equally distribute the requests across all HN-F and SN-F target devices. By programming select bits in the rnsam_hash_addr_mask_reg address mask register to `0b0`, those address bits can be removed from the hashing logic. This feature is only applicable to hashed memory regions.

The following limitations apply:

- 3-SN, 5-SN, and 6-SN mode is enabled, a configurable group of nine address bits in the range of address bits [21:8] and the configured top_addr_bits are essential in distributing the addresses between memory. Arm recommends that these bits are masked carefully to avoid memory aliasing. See 3.4.7.1 HN-F to SN-F memory striping in HN-F SAM on page 152
- Range compare mask must not mask off bits that represent the size of the region. For example, if any of the region sizes are 64MB, address bit 26 cannot be masked. Similarly, if a region size is 512MB, address bit 29 cannot be masked.
- The address bits masked in HN-F and RNSAM must be consistent. This requirement ensures that PrefetchTarget requests from an RN-F to SN-F are addresses by the same SN-F as SLC miss from the HN-F to SN-F.

Table 3-35: RN SAM parameters and configuration registers

RN SAM feature	Applicable parameters	Applicable configuration registers
Global	RNSAM_FLEX_TGTID_EN	rnsam_status rnsam_region_cmp_addr_mask_reg
GIC region	-	gic_mem_region_reg
Non-hashed regions	RNSAM_NUM_NONHASH_REGION	non_hash_mem_region_reg0-63 non_hash_mem_region_cfg2_reg0-63 non_hash_tgt_nodeid0-15
	RNSAM_NONHASH_RCOMP_EN	
	RNSAM_NONHASH_RCOMP_LSB	

RN SAM feature	Applicable parameters	Applicable configuration registers
Hashed regions (HN-F)	RNSAM_NUM_HTG	rnsam_hash_addr_mask_reg
	RNSAM_NP2_EN	sys_cache_grp_region0-3
	RNSAM_HIER_HASH_EN	hashed_tgt_grp_cfg1_region4-31
	RNSAM_HTG_RCOMP_EN	hashed_tgt_grp_cfg2_region0-31
	RNSAM_HTG_RCOMP_LSB	sys_cache_grp_secondary_reg0-3
	RNSAM_NUM_ADD_HASHED_TGT	hashed_target_grp_secondary_cfg1_reg4-31
		hashed_target_grp_secondary_cfg2_reg0-31
		sys_cache_group_hn_count
		hashed_target_group_hn_count_reg1-3
		sys_cache_grp_nonhash_nodeid
		hashed_target_grp_nonhash_nodeid_reg1-6
		sys_cache_grp_hn_nodeid_reg0-15
		hashed_target_grp_hnf_nodeid_reg16-31
		sys_cache_grp_cal_mode_reg
		hashed_target_grp_cal_mode_reg1-7
		hashed_target_grp_hash_cntl_reg0-31
		hashed_target_grp_hnf_device_bound_cfg_reg0-1
Hashed regions (HN-P/CCG)	RNSAM_AXID_HASH_EN	rnsam_hash_axi_id_mask_reg
		hashed_target_grp_hnp_nodeid_reg0-15 hashed_target_grp_hnp_cpa_en_reg0
QoS regions	RNSAM_NUM_QOS_REGIONS	sam_qos_mem_region_reg0-7
		sam_qos_mem_region_cfg2_reg0-7

RN SAM feature	Applicable parameters	Applicable configuration registers
CPAG support	RNSAM_NUM_CPA_GRP	sys_cache_grp_hn_cpa_en_reg hashed_target_grp_hnf_cpa_en_reg1 sys_cache_grp_hn_cpa_grp_reg hashed_target_grp_cpa_grp_reg1-7 cml_port_aggr_mode_ctrl_reg cml_port_aggr_mode_ctrl_reg1-6 cml_port_aggr_grp0-31_add_mask cml_port_aggr_grp_reg0-12 cml_port_aggr_ctrl_reg cml_port_aggr_ctrl_reg1-6
PrefetchTgt support	RNSAM_PREFETCH_EN RNSAM_PFTGT_NUM_SCG RNSAM_PFTGT_NUM_NONHASH_PSCG RNSAM_PFTGT_NUM_HTG_PSCG RNSAM_PFTGT_DEF_HASHED_GRP_EN	sys_cache_grp_sn_nodeid_reg0-31 sys_cache_grp_hashed_regions_sn_nodeid _reg0-15 sys_cache_grp_sn_attr sys_cache_grp_sn_attr1 sys_cache_grp_sn_sam_cfg0-3 sys_cache_grp_region0_sn_nodeid_reg0-31 sys_cache_grp_region1_sn_nodeid_reg0-31 sam_scg0-7_prefetch_nonhashed_mem_region _cfg1_reg0-63 sam_scg0-7_prefetch_nonhashed_mem_region _cfg2_reg0-63 sam_scg0-7_prefetch_hashed_region_cfg1_reg0-7 sam_scg0-7_prefetch_hashed_region_cfg2_reg0-7 sam_scg0-7_prefetch_hashed_region_cfg3_reg0-7
Custom hash	RNSAM_CUSTOM_REG	sam_generic_regs0-7

Note

- Parameters and registers in the same row are not necessarily related
- Ranges in the register names, m-n, indicates the register repeats (m-n+1) times
- Register names with two ranges, k-l and m-n, indicates the register repeats (l - k + 1) × (m - n + 1) times

3.4.6.10 RN SAM Compact HN Tables for multi-chip configurations

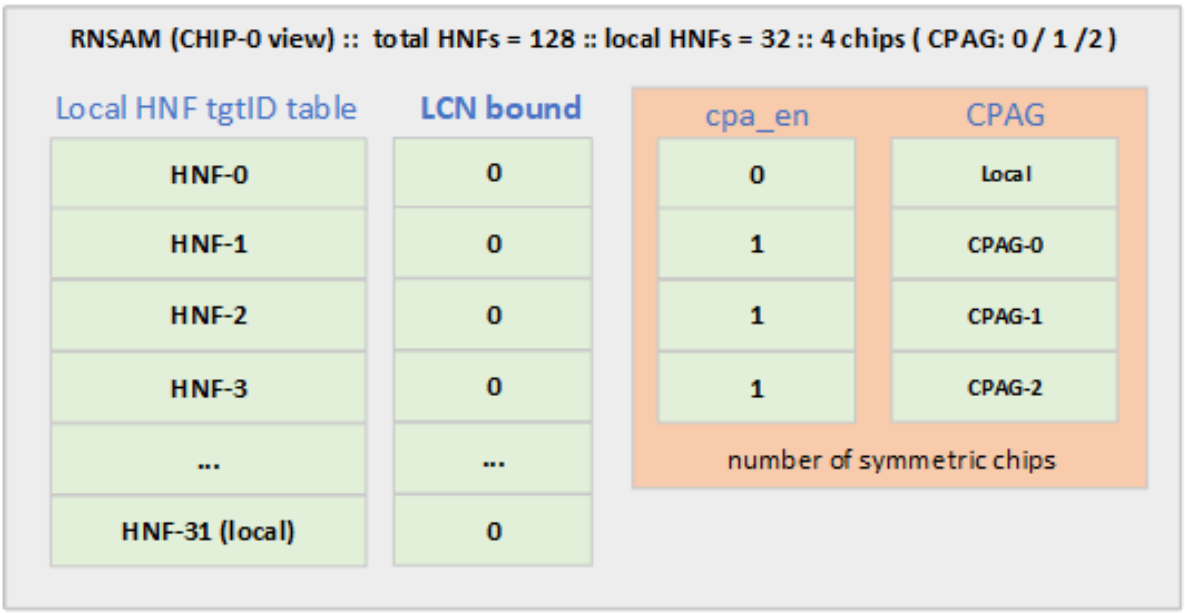
The CMN-700 multi-chip configurations with single NUMA, hashing across local and remote HN-Fs RN SAM logic requires only local HN-F target-ids to be programmed. The RN SAM targets CCG node-ids for the remote HN-F target-ids.

Considering only local HN-F target-ids you can build the SAM with minimal flops.

This mode is enabled by setting the parameter `COMPACT_HN_TABLES_EN_PARAM`. The Compact HN tables mode is contrary to the legacy modes where SAM build flops for both the local and remote HN-Fs. Compact HN tables mode is limited only to multi-chip systems with the same HN counts across all chips. The local HN-F target-id table look up bits are derived from the total HN-F hash selection bits. The following figure shows an example of the Compact HN table based on a configuration with 128 HNFs and 4 symmetric chips.

For Compact HN tables mode only power-of-two and Hierarchical hashing modes are supported.

Figure 3-49: Compact HN tables



The local HN-F and CPA Group hash index is determined by shuttering the bits from total HN-F hash index bits based on the configuration. For example the SAM performs a hash for the entire 128 HNFs = 7bit hash index bits [6:0]:

- CPAG hash index = total HN-F hash index [1:0] = 4 symmetrical chips = 2 bits
- Local HN-F hash index = total HN-F hash index [6:2] = 32 local HN-Fs = 5 bits

For Hierarchical hashing, the shuttering of hash bits only applies to the cluster hash (first hash). For example the SAM performs hierarchical hashing across 128 HN-Fs (16 clusters x 8 HN-Fs per cluster)

The cluster hash index (first hash) = [3:0] (16 cluster)

HN-F nodes hashes the index within a cluster = [2:0] (8 nodes per cluster)

- CPAG hash index = cluster hash index [1:0] = 4 symmetrical chips = 2 bits
- Local HN-F hash index = cluster hash index [3:2] = 4 cluster per chip = 32 local HN-F

All the HN-Fs within a cluster should be within one chip.

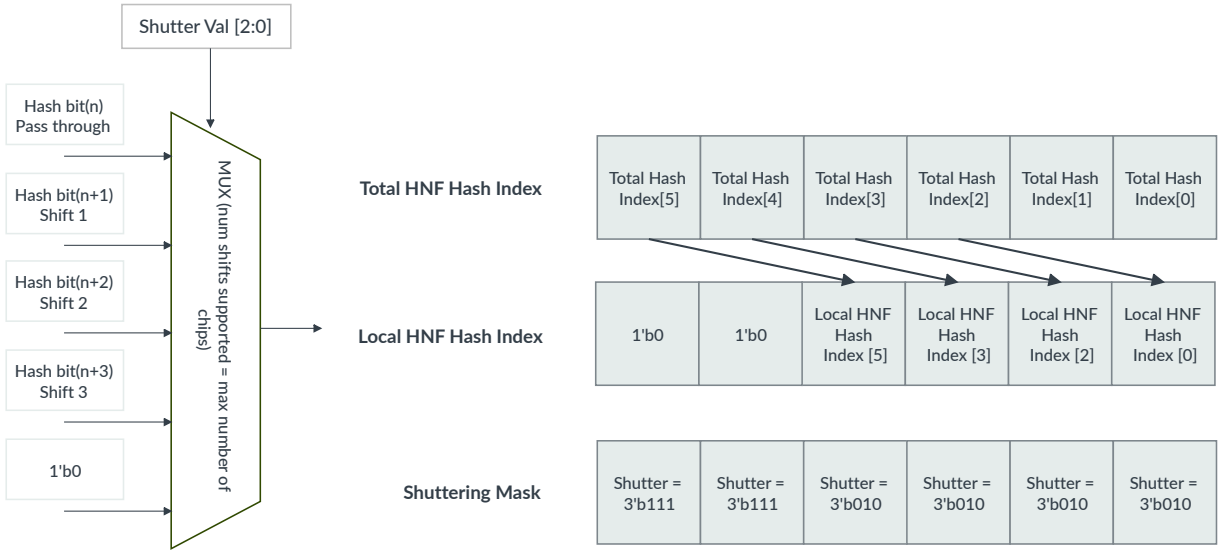
3.4.6.10.1 Local HN-F and CPAG hash index: shuttering from total hash index bits

To configure shuttering value for each Local HN-F hash index and CPAG hash index.

- Total HN-F hash index bits: Supports maximum of 128 HN-Fs (256 HN-F in CAL2) - 7 bits
- Total HN-F hash index bits: Supports maximum of 64 HN-Fs per chip (128 HN-F in CAL2) - 6 bits
- CPAG hash index bits: Supports maximum of 32 symmetrical chips – 5bits

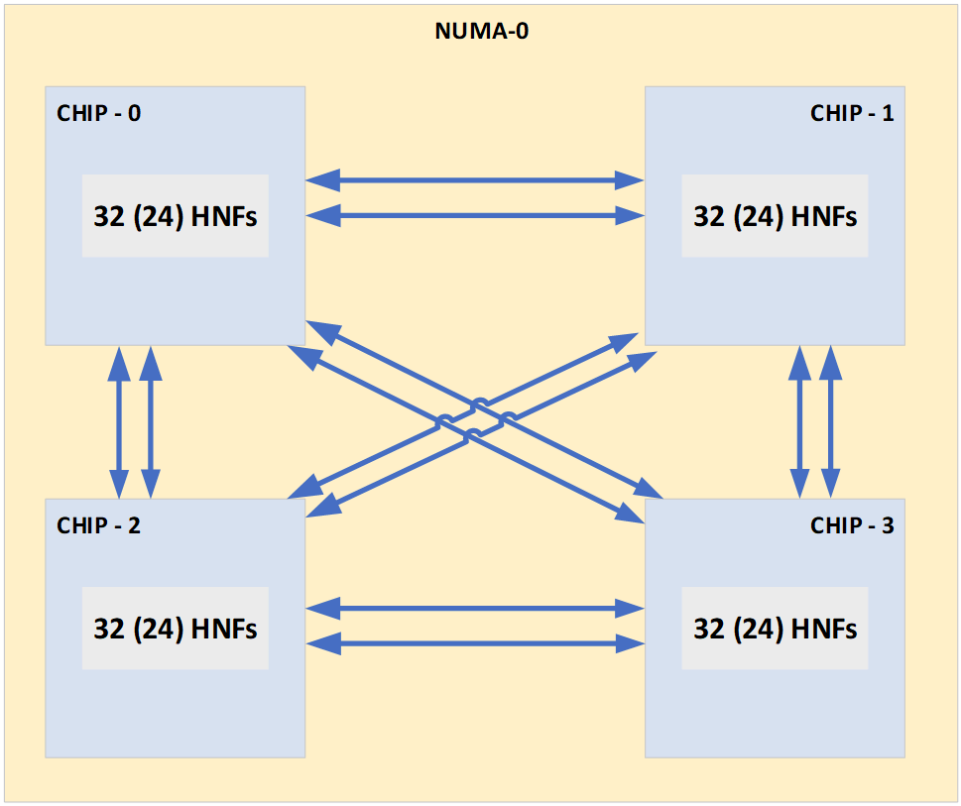
The SAM uses programming structures to determine HN-F selection, where hash bits are selectively picked from the overall hash bit, as shows the following figure:

Figure 3-50: Configure shuttering value



The following figure shows a programming of a four Chip SMP in a single NUMA configuration, which includes a power of two HN-Fs and non-power of two HN-Fs, using 24 as an example.

Figure 3-51: 4 CHIP SMP (1 NUMA)



The following table shows the programmed configuration fields for a power of two HN-F Hashed Target Group for the previous single NUMA figure.

Table 3-36: Single Numa power of two Hashed Target Group configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	128	hnf_hash[6:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hns_sel_shuttering[6:0][3:0]	{value}	hnf_hash[6:2]
cpa_grpid_shuttering[4:0][3:0]	{value}	hnf_hash[1:0]
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hnf_base_index	0	-

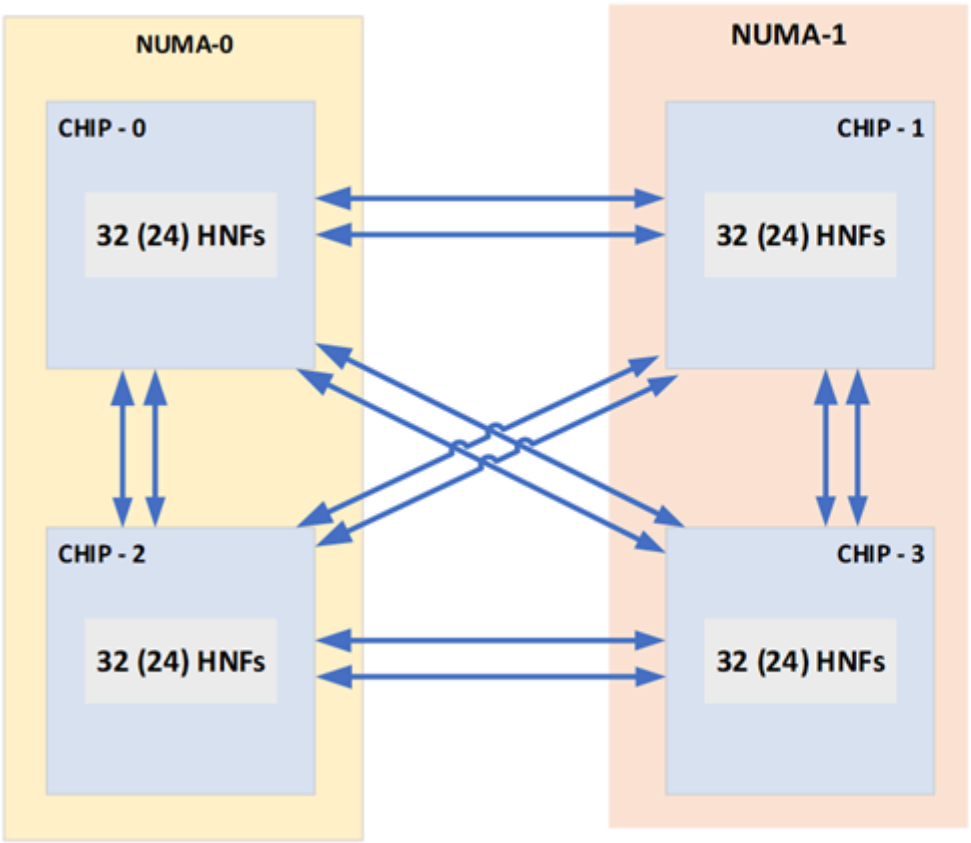
The following table shows the programmed configuration fields for a non-power of two HN-F Hashed Target Group for the previous single NUMA figure.

Table 3-37: Single Numa non-power of two Hashed Target Group configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	96	-
hierarchical_hash_en	1	-
hier_hash_cluster	8	clstr_hash[2:0]
hier_hash_nodes	12	node_hash[3:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hier_addr_shuttering[51:6][2:0]	{value}	addr[51:9]
hns_sel_shuttering[6:0][3:0]	{value}	clstr_hash[2] x node_hash[3:0]
cpa_grpid_shuttering[4:0][3:0]	{value}	hnf_hash[1:0]
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hnf_base_index	0	-

The following figure shows a programming of a Multi Chip SMP in a two NUMA configuration, which includes a power of two HN-Fs and non-power of two HN-Fs using 12 as an example.

Figure 3-52: Multi CHIP SMP (2 NUMA)



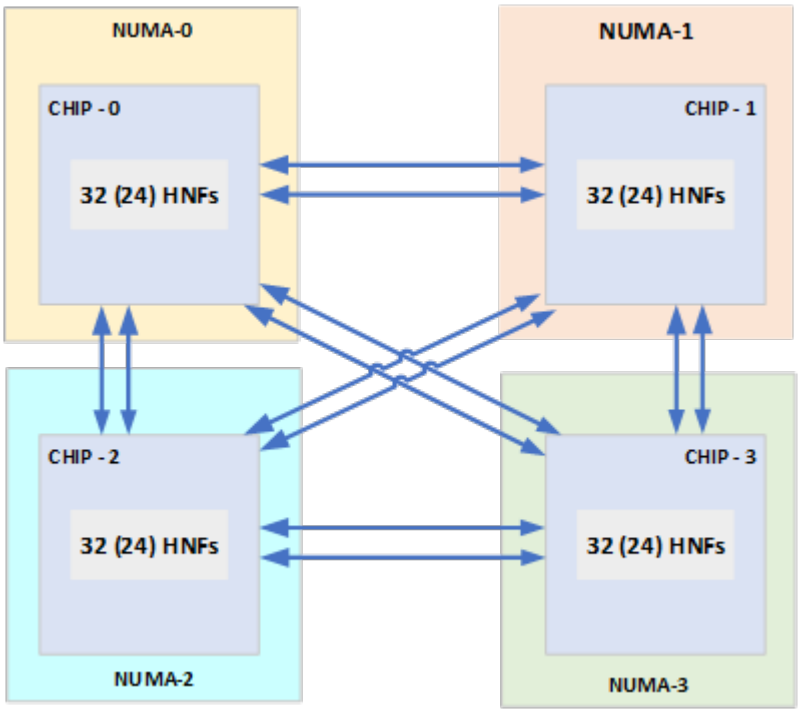
The following table shows the programmed configuration fields for a power of two HN-F Hashed Target Group for the previous two NUMA figure.

Table 3-38: Two Numa power of two Hashed Target Group[0/1] configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	64	hnf_hash[5:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hns_sel_shuttering[6:0][3:0]	{value}	hnf_hash[5:2]
cpa_grpid_shuttering[4:0][3:0]	{value}	hnf_hash[1:0]
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hnf_base_index	0	-

The following figure shows a programming of a four Chip SMP in a four NUMA configuration, which includes a power of two HN-Fs and non-power of two HN-Fs using 24 as an example.

Figure 3-53: 4 CHIP SMP (4 NUMA)



The following table shows the programmed configuration fields for a power of two HN-F Hashed Target Group for the previous four NUMA figure.

Table 3-39: Two Numa power of two Hashed Target Group[0/1/2/3] configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	64	hnf_hash[5:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hns_sel_shuttering[6:0][3:0]	{value}	hnf_hash[4:0]
cpa_grpid_shuttering[4:0][3:0]	{value}	always point to index - 0
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hnf_base_index	0	-

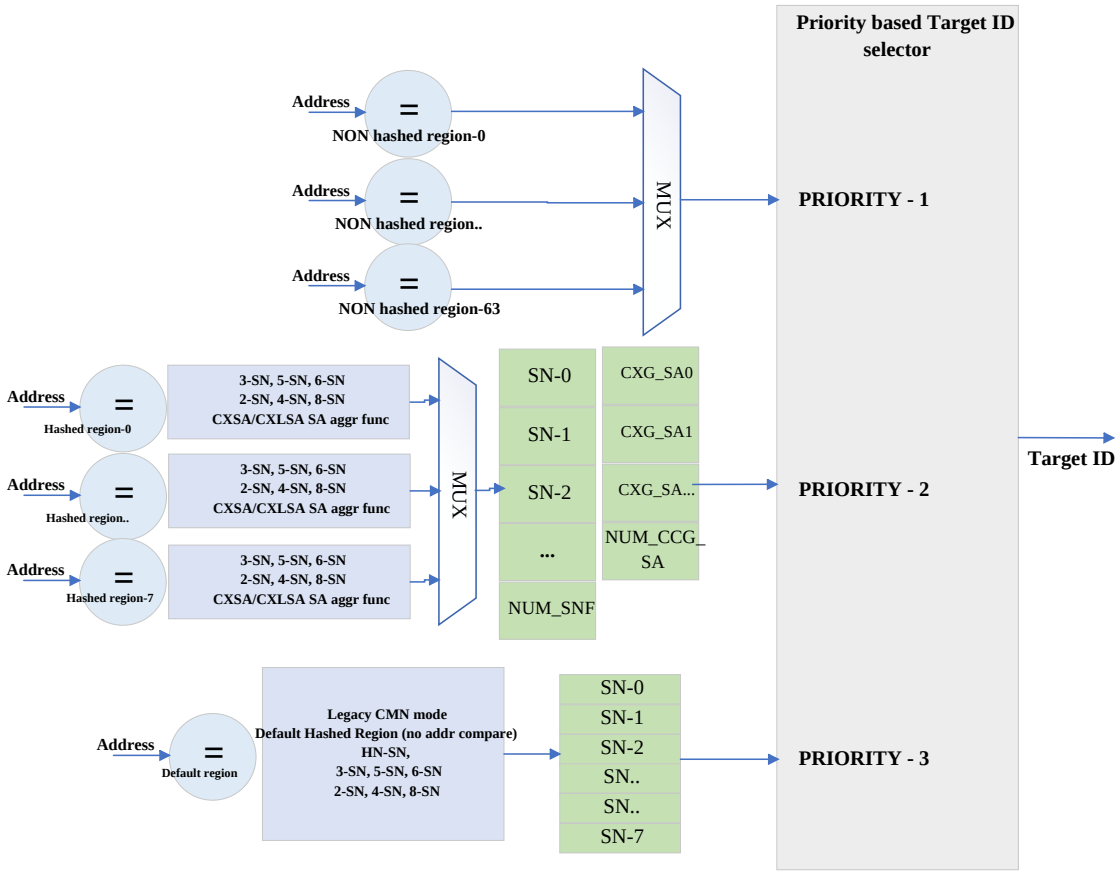
3.4.7 HN-F SAM

Transactions from an HN-F to an SN must pass through an HN-F SAM to generate a CHI target ID.

4.4.3 RN SAM and HN-F SAM programming on page 1121 describes the HN-F SAM programming sequence.

The following figure shows how the target ID is determined from the HN-F SAM.

Figure 3-54: HN-F SAM target ID selection policy



The preceding figure shows that the HN-F SAM has the following mapping policies to generate target IDs for transactions:

- Range-based:
  - Non-hashed
  - Hashed target groups
- Default hashed region (Legacy CMN mode):
  - Hashed
  - Direct mapping

An HN-F cannot use both hashed mapping mode and direct mapping mode. To map transactions that fall in the range-based part of the HN-F SAM, HN-F SAMs support priority-based target ID selection. The order of priority when selecting a target ID is:

1. Range-based: non-hashed SN target ID
2. Range-based: hashed SN target ID
3. Default Hashed region: Hashed target ID or direct mapped target ID (Legacy CMN mode).

HNSAM range-based regions support two ways of defining a memory region and it is selected through a user parameter HNSAM_RCOMP_EN.

- Base address & region size, size of the region being power of two (64MB minimum region size; HNSAM_RCOMP_EN = 0).
- Lower address & upper address, no restrictions on size of the region. (HNSAM_RCOMP_EN = 1)

Minimum size of the memory region is selected through a user parameter (HNSAM_RCOMP_LSB = [20-26]) that defines the lower address bit for the comparison. This is limited to the lower address & upper address mode only.

- HNSAM_RCOMP_LSB = 20, defines minimum memory size = 1MB
- HNSAM_RCOMP_LSB = 21, defines minimum memory size = 2MB
- HNSAM_RCOMP_LSB = 22, defines minimum memory size = 4MB
- HNSAM_RCOMP_LSB = 26, defines minimum memory size = 64MB

Range-based mapping: Non-hashed target IDs

Non-hashed range-based mapping is an address-based unique target ID generation policy. Up to 64 memory regions (HNSAM_NUM_NONHASH <= 64) can be created, each targeting a single SN. This mode is useful where a partition of memory from the global DRAM is mapped explicitly to an individual SN, for example, an on-chip SRAM.

Range-based mapping: Hashed target groups

Hashed range-based mapping is an address based hashed target ID generation policy. Up to eight memory regions can be created, each targeting a group of SNs. This mode is useful when a partition of memory from the global DRAM space is mapped explicitly to a group of SNs. These modes can only be used when a DRAM partition targets the following SN configurations:

- 3-SN mode: Addresses from a hashed region are striped across three SNs
- 5-SN mode: Addresses from a hashed region are striped across five SNs
- 6-SN mode: Addresses from a hashed region are striped across six SNs
- 2-SN mode: Addresses from a hashed region are striped across two SNs
- 4-SN mode: Addresses from a hashed region are striped across four SNs
- 8-SN mode: Addresses from a hashed region are striped across eight SNs

In 3-SN, 5-SN, or 6-SN mode, addresses are striped at a 256MB granularity between the 3, 5, and 6 SNs. The stripe function uses a configurable group of nine address bits in the range of address bits [21:8] and an extra two (3-SN) or three (5-SN, 6-SN) user-defined address bits.

In 2-SN, 4-SN, or 8-SN mode, addresses are striped at a 64B granularity between the 2SNs/4SNs/8SNs. The stripe function uses XOR of address bits [MSB:6].

- Two SNs:
  - Number of bits in select: 1
  - Select [0] = (6^7^8^ ...^51)

- Four SNs:
  - Number of bits in select: 2
  - Select [0] = (6^8^10^ ...^50)
  - Select [1] = (7^9^11^ ...^51)
- Eight SNs:
  - Number of bits in select: 3
  - Select [0] = (6^9^12^ ...^51)
  - Select [1] = (7^10^13^ ...^49)
  - Select [2] = (8^11^14^ ...^50)

Default Hashed region: Hashed mapping (legacy CMN mode)

The HN-F SAM supports the following hashed modes for SN target ID selection. The default hashed region does not use address comparison. When an incoming address does not match any range-based mapping, then SN TargetID is derived from the default hashed region:

2-SN mode
: Addresses from a given HN-F are striped across two SNs

3-SN mode
: Addresses from a given HN-F are striped across three SNs

4-SN mode
: Addresses from a given HN-F are striped across four SNs

5-SN mode
: Addresses from a given HN-F are striped across five SNs

6-SN mode
: Addresses from a given HN-F are striped across six SNs

8-SN mode
: Addresses from a given HN-F are striped across eight SNs

The default hashed region has address comparison logic added to detect for incorrect address region access. The default hashed regions address boundaries must match the RNSAM HTG address boundaries.

The HNSAM notifies the HN-F upon an address compare miss if the address does not match any programmed SAM ranges.

The HN-F does not generate NDE (non-data error) responses on decode error.

- Write response: The HN-F will drop the response
- Read response: The HN-F sends the response back to the RN-F or CCG with poison.

To maintain backwards compatibility, program the base_addr = 0x0 and size = 0xFF or end_addr = 0xFFFFFFFFF, depending on which RCOMP parameter is used.

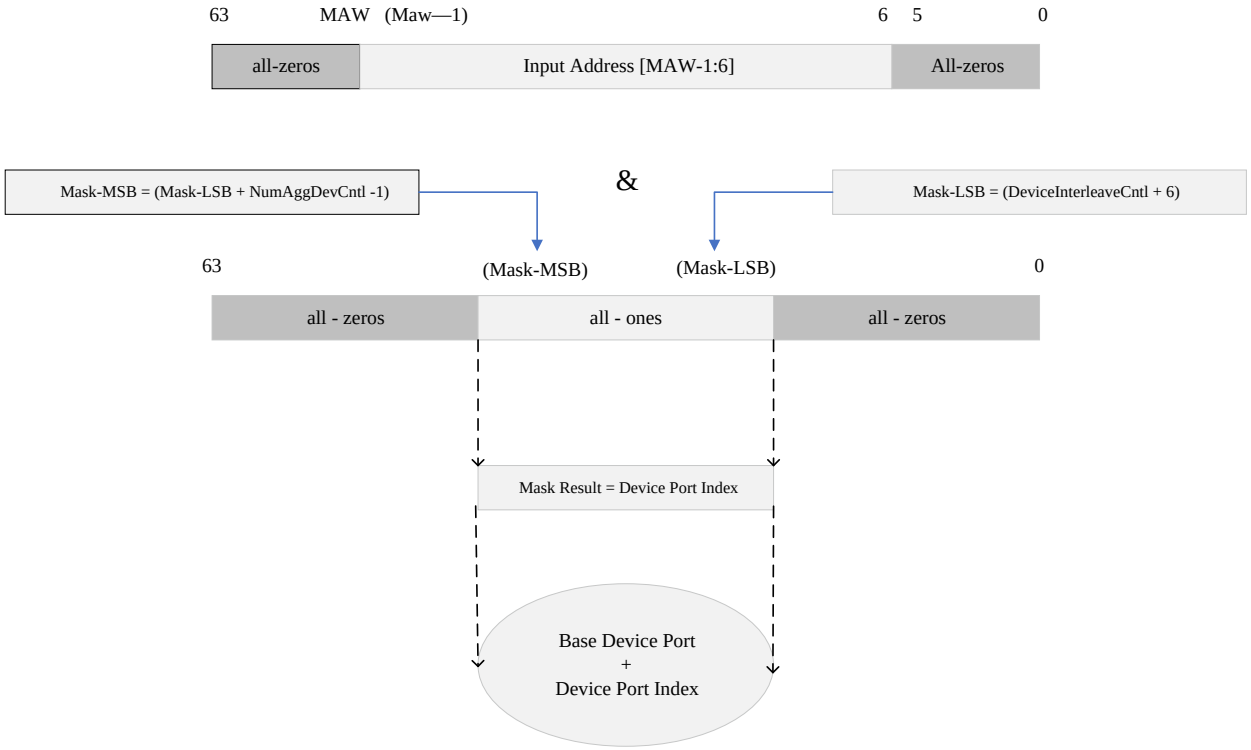
Direct mapping

Direct SN mapping is used if the SCG targets 1, 2, 4, 8, 16, 32 SNs. When attached using CAL, 64 SNs are supported. In this case, the transaction uses the SN0 target ID. Distributing accesses across the SNs targeted by the SCG is achieved by programming the SN0 field of each HN-F to the SN node ID it targets. For example, if an SCG with eight HN-Fs targets eight SNs, the SN0 field of each HN-F would be programmed with a different SN node ID. If that same SCG with eight HN-Fs targets four SNs instead, every two HN-F nodes would have the same SN0 field value.

Aggregated SA selection function

HNSAM supports hashing across multiple CXL.MEM devices.

Figure 3-55: Aggregated SA selection function



Aggregated SA selection requires the following configurations to determine the SN target ID:

- Device interleaving: This configuration controls the interleave size across all aggregated CXSA devices and supports interleave sizes between 64 Bytes to 2Mbytes. This controls the number of LSB bits to be stripped from the address.
- Aggregated devices count: This configuration controls the number of the CXSA devices aggregated. The number of aggregated devices supported are 1, 2, 4, 8, and 16. This controls the number of MSB bits to be stripped from the address.

- HNSAM supports maximum of 16 CXSA devices in the HNSAM HTG regions (CXSA port aggregation). Additional CXSA devices need to be supported through HNSAM non-hashed regions.

Address bits extracted by stripping the LSB and MSB bits based on the above configuration yields the selection bits for deriving the CXSA target ID.

User defined hashing logic

The hash modules can be selectively replaced to implement user-defined hash logic. HN SAM implements user-defined configuration registers to be used for the user-defined hash logic. The number of user-defined registers are enabled using a user parameter (`HNSAM_CUSTOM_REGS [0-2]`).

For more information about modifying the hash modules, see *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*.

SN Target-ID tables

HNSAM hashed target groups use the Target-IDs from the Target-ID structure with depth determined by number of SNs in the configuration. Maximum SN Target ID entries supported are 64.

HNSAM hashed target groups use CXSA Target IDs from the separate Target ID structure with depth determined by number of remote SNs in the configuration. The maximum CCG SA target ID entries supported are 16.

Each hashed target group generates the index based on the address striping and lookup into the target ID tables for the SN and CCG target ID. Base index for each HTG is derived based on the linked list mechanism from the boot time programming.

Hashed region[n] base index = (Hashed region [n-1] base index) + (Hashed region[n-1] num SN)

3.4.7.1 HN-F to SN-F memory striping in HN-F SAM

The CMN-700 HN-F SAM supports three non-power-of-two memory striping modes, which are known as 3-SN mode, 5-SN mode, and 6-SN mode respectively. In these modes, the HN-F stripes addresses across three SN-Fs, five SN-Fs, or six SN-Fs respectively.

All three striping modes use a configurable group of nine address bits in the range of address bits [21:8] and extra upper address bits configured as top_addr_bits as inputs to their stripe function. The address bit group used can be one of: [16:8] (Default), [17:9], [18:10], [19:11], [20:12], or [21:13].

3-SN mode

In 3-SN mode, a stripe function ensures that traffic is distributed evenly among the three SNs. The two higher PA bits are referred to as top_address_bit1 and top_address_bit0. The top address bits are selected so that three of the four combinations of the top address bits appear evenly in the selected address space, and the fourth combination never appears.


Note

In some situations, a top bit can be the inverse of the selected PA bit.

For each physical address, one of the three SNs is selected using the following formula when address bits [16:8] are used (default):

SN = { ADDR[10:8] + ADDR[13:11] + ADDR[16:14] + ((top_addr_bit1<<1) | top_addr_bit0) } % 3

General SN distribution behavior example

For a simple case with a 3GB flat address space starting at address `0x0`, top_address_bit1 is PA[31], and top_address_bit0 is PA[30]. With increasing physical address, the function steps between SNs at a 256-byte granularity. As the physical address iterates from 0-128KB, with top_address_bit1 = top_address_bit0 = 0, the first three terms distribute the traffic relatively evenly among the SNs. Of the 512 blocks, 256B each, in the first 128KB, the distribution is:

SN[0] 170 blocks 33.2%
SN[1] 171 blocks 33.4%
SN[2] 171 blocks 33.4%

This pattern repeats over each 128KB until 1GB, where top_address_bit0 toggles. With top_address_bit1 = 0 and top_address_bit0 = 1, the pattern is shifted. For each 128KB:

SN[0] 171 blocks 33.4%
SN[1] 170 blocks 33.2%
SN[2] 171 blocks 33.4%

At 2GB, when top_address_bit1 = 1 and top_address_bit0 = 0, the pattern shifts again:

SN[0] 171 blocks 33.4%
SN[1] 171 blocks 33.4%
SN[2] 170 blocks 33.2%

Over the full 3GB, the same number of lines are distributed to each SN.

The HN-F uses the hn_cfg_three_sn_en bit in its cmn_hns_sam_control register to enable routing to three SNs. In the cmn_hns_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 two address bits are decoded, and used with a hashing function to determine the target SN-F.

5-SN and 6-SN modes

Similar to 3-SN hashing, the 5-SN and 6-SN modes extend the function to equally distribute addresses between five or six SNs respectively. For each physical address, one of the three SNs is selected using the following formula when address bits [16:8] are used (default):

SN = { ADDR[10:8] + ADDR[13:11] + ADDR[16:14] + ((top_addr_bit2<<2) | (top_addr_bit1<<1) | top_addr_bit0) } % 6

HN-F SAM uses the following bits in the cmn_hns_sam_control register to enable striping across five or six SN-Fs:

- hn_cfg_five_sn_en
- hn_cfg_six_sn_en

In 5-SN and 6-SN mode, HN-F SAM also uses hn_cfg_sam_top_address_bit2 field in the cmn_hns_sam_control register along with hn_cfg_sam_top_address_bit1 and hn_cfg_sam_top_address_bit0 to hash the incoming address.

3-SN, 5-SN, and 6-SN configurations


Note

In each of the following examples, address bits [16:8] are used in the stripe function (the default).

This configuration ensures equal distribution of requests across all SN-Fs and prevents memory aliasing. The SAM also provides an inv_top_address_bit configuration bit, which can be used with top address bits. The following table shows the valid top address bits for Arm PDD Memory Map.

See the *Principles of Arm® Memory Maps White Paper*.

Table 3-40: 3-SN mode top address bits[bit 1, bit 0]

Combination 1 (inv_top_address_bit set to 0b0)	Combination 2 (inv_top_address_bit set to 0b0)	Combination 3 (inv_top_address_bit set to 0b1)	Combination 4 (inv_top_address_bit set to 0b1)
[0, 0]	[1, 1]	[0, 0]	[0, 0]
[0, 1]	[0, 1]	[1, 0]	[0, 1]
[1, 0]	[1, 0]	[1, 1]	[1, 1]


Note

When inv_top_address_bit=1, it forces the SAM to invert the top most significant top address bit. For 3-SN mode, top_address_bit1 is inverted. For 5-SN and 6-SN mode, top_address_bit2 is inverted.

The following table shows the valid combinations for the address bits for 6-SN mode with PDD memory map.

Table 3-41: 6-SN mode top address bits[bit 2, bit 1, bit 0]

Combination 1 (inv_top_address_bit set to 0b0)	Combination 2 (inv_top_address_bit set to 0b0)	Combination 3 (inv_top_address_bit set to 0b1)
[0, 0, 0]	[0, 1, 0]	[0, 0, 0]
[0, 0, 1]	[0, 1, 1]	[0, 0, 1]

Combination 1 (inv_top_address_bit set to 0b0)	Combination 2 (inv_top_address_bit set to 0b0)	Combination 3 (inv_top_address_bit set to 0b1)
[0, 1, 0]	[1, 0, 0]	[0, 1, 0]
[0, 1, 1]	[1, 0, 1]	[0, 1, 1]
[1, 0, 0]	[1, 1, 0]	[1, 1, 0]
[1, 0, 1]	[1, 1, 1]	[1, 1, 1]

The combinations for 5-SN mode must follow similar rules to 6-SN mode programming. The top address bit combinations must be five sequential combinations from the preceding table.

Example 3-6: Example for PDD memory map

Assume that a system supports three SN-Fs with 32GB of DRAM at each SN-F port and all three SN-Fs are used for SCG 0. The first two regions together comprise a 32GB region, while the remaining two regions are 32GB each. Because the DRAM space is non-contiguous in this memory map (2GB + 30GB + 480GB), the base addresses for each DRAM partition are:

1. a. `000_8000_0000` to `000_FFFF_FFFF` (2GB)
   b. `008_8000_0000` to `00F_FFFF_FFFF` (30GB)
2. `088_0000_0000` to `08F_FFFF_FFFF` (32GB)
3. `090_0000_0000` to `097_FFFF_FFFF` (32GB)

The following table breaks down the address bits for the regions that the previous list shows.

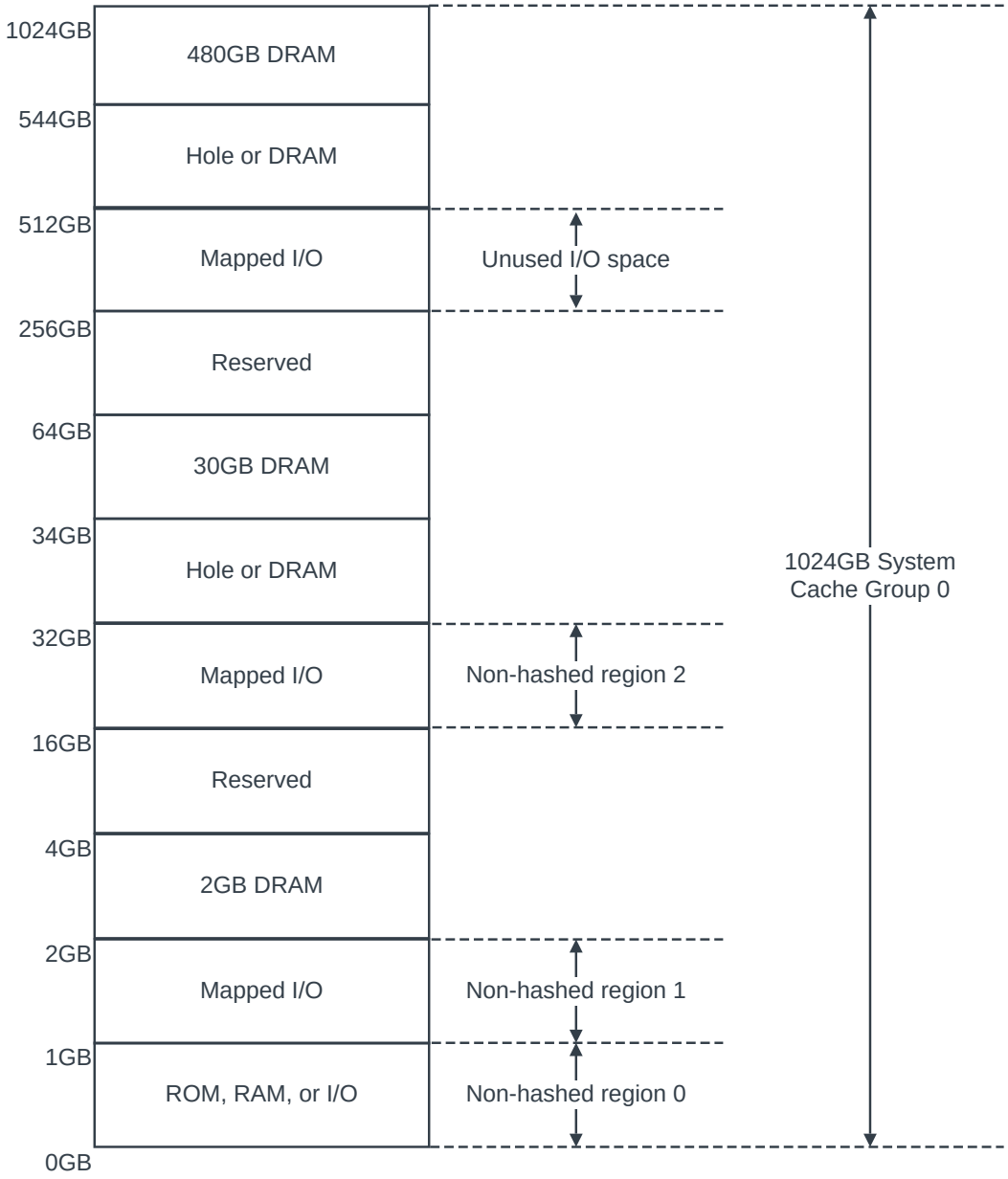
Table 3-42: Address region example bit settings: 3-SN example

Region	39	38	37	36	35	34	33	32	31
1a	0	0	0	0	0	0	0	0	1
1b	0	0	0	0	1	x	x	x	1
2	1	0	0	0	1	x	x	x	x
3	1	0	0	1	0	x	x	x	x

From the address bit breakdown, the selected top address bits must make sure both regions that are marked as Region 1 have the same values. This requirement ensures no aliasing in memory. For this memory map and DRAM size, there are no address bits that directly give Combination 1 or Combination 2 as shown in Table 3-40: 3-SN mode top address bits[bit 1, bit 0] on page 154. However, if bits[39, 36] are used along with inv_top_address_bit = 1, then Combination 3 is possible. This approach ensures that the memory requests are equally distributed across the three SN-Fs without memory aliasing.

The following figure shows the Arm proposed memory map.

Figure 3-56: Example memory map programming



The following tables provide example address bits that provide equal distribution of memory across all SN-Fs in 3-SN and 6-SN hashed modes.

The following shows the 3-SN DRAM size settings.

Table 3-43: 3-SN DRAM size settings

3-SN DRAM size at each SN-F port	Top address bits[bit 1, bit 0]	Inv_top_address_bit value
1GB, total 3GB	[35, 30]	0b0

3-SN DRAM size at each SN-F port	Top address bits[bit 1, bit 0]	Inv_top_address_bit value
2GB, total 6GB	[35, 31]	0b0
4GB, total 12GB	[33, 32]	0b0
8GB, total 24GB	[34, 33]	0b0
16GB, total 48GB	[39, 34]	0b0
32GB, total 96GB	[39, 36]	0b1
64GB, total 192GB	[37, 36]	0b0
128GB, total 384GB	[38, 37]	0b0

The following shows the 6-SN DRAM size settings.

Table 3-44: 6-SN DRAM size settings

6-SN DRAM size at each SN-F port	Top address bits[bit 2, bit 1, bit 0]	Inv_top_address_bit value
1GB, total 6GB	[35, 31, 28]	0b0
2GB, total 12GB	[33, 32, 28]	0b0
4GB, total 24GB	[34, 33, 28]	0b0
8GB, total 48GB	[39, 34, 33]	0b0
16GB, total 96GB	[39, 36, 28]	0b1
32GB, total 192GB	[37, 36, 28]	0b0
64GB, total 384GB	[38, 37, 28]	0b0

The top address bit combinations for 5-SN mode are not available with the PDD memory map for all DDR size combinations. For specific memory maps, you must follow the 6-SN rules to achieve valid address bit combinations and contiguous SN addresses.

3.4.7.2 SN contiguous address spaces

The following describes which physical address bits must be connected to an SN-F for various configurations.

If all HN-Fs send their cache misses to a single SN-F, that SN-F sees the full address space. However, it is common for a system to have two or more SN-Fs, with each HN-F sending its cache misses to a single SN-F. In this scenario, each SN-F receives only part of the address space. SN-F typically removes one or more address bits to retain a contiguous address map. The full physical address is presented to each SN-F for every request so that any SN-F based memory protection logic can function. However, the actual mapping to RAM locations can be done with the modified address. The address modification depends on multiple factors:

- Number of HN-Fs in the cache group
- Number of SN-Fs in the cache group
- Which HN-Fs share SN-Fs

2^n-SN address striping

The following table provides HN-F and SN-F combinations that are supported within a cache group, along with the address bits that should be removed.

Table 3-45: HN-F and SN-F combinations supported within a cache group

Number of HN-Fs	Number of SN-Fs	Bits to strip from full PA
2	1	None
	2	[6]
4	1	None
	2	[7]
	4	[7, 6]
8	1	None
	2	[8]
	4	[8, 7]
	8	[8, 7, 6]
16	1	None
	2	[9]
	4	[9, 8]
	8	[9, 8, 7]
	16	[9, 8, 7, 6]
32	1	None
	2	[10]
	4	[10, 9]
	8	[10, 9, 8]
	16	[10, 9, 8, 7]
	32	[10, 9, 8, 7, 6]
64	1	None
	2	[11]
	4	[11, 10]
	8	[11, 10, 9]
	16	[11, 10, 9, 8]
	32	[11, 10, 9, 8, 7]

The method that is used to calculate the bits stripped is as follows:

1. The highest bit removed is the least significant address bit used in the XOR function for the most significant bit of the HN-F select hash function.

   256 HN-Fs

   PA[13]

   128 HN-Fs

   PA[12]

64 HN-Fs
PA[11]

32 HN-Fs
PA[10]

16 HN-Fs
PA[9]

Eight HN-Fs
PA[8]

Four HN-Fs
PA[7]

Two HN-Fs
PA[6]

2. The number of bits stripped is $\log_2$(number of SN-Fs), sequentially below the highest bit.

This approach to bit stripping assumes that HN-Fs that share SN-Fs are sequential in the RN SAM cache group HN-F table. For example, if there are eight HN-Fs and two SN-Fs, the bottom four HN-Fs in the RN SAM table would share an SN-F. The top four HN-Fs would share an SN-F.

3-SN, 5-SN, and 6-SN address striping

As 3-SN, 5-SN, and 6-SN address hashing implements modulo function according to the top address bits used, the SN-F must remove these bits to achieve a contiguous memory map in the DRAM.

3-SN mode	The SN-F must remove top_address_bit1 and top_address_bit0.
5-SN mode	The SN-F must remove top_address_bit2, top_address_bit1, and top_address_bit0.
6-SN mode	The SN-F must remove top_address_bit2, top_address_bit1, and top_address_bit0.

3.4.7.3 Address bit masking in the HN-F SAM

CMN-700 supports masking of address bits in the HN-F SAM. Certain restrictions apply to this process.

Range-based comparison

HN-F SAM uses address bits:

- [MSB:26] when the applicable `*RCOMP_EN` parameter is 0.
- [MSB:2`*RCOMP_LSB`] When the applicable `*RCOMP_EN` parameter is 1

By programming select bits to `0b0` in the hnf_sam_region_cmp_addr_mask_reg mask register, the incoming address and the programmed address ranges can be masked off before comparison. This region mask is only applicable to region-based memory partitioning in the HN-F and so the mask is not applied to the hashing scheme in 3-SN, 5-SN, and 6-SN modes.

Stripe function

HN-F SAM supports masking of address bits used for 3-SN, 5-SN, or 6-SN address hashing. This feature can be enabled by programming the hn_sam_hash_addr_mask_reg.

The following limitations apply:

- Range compare mask must not mask off bits that represent the size of the region. For example, if any of the region sizes are 64MB, address bit 26 cannot be masked. Similarly, if a region size is 512MB, address bit 29 cannot be masked.
- If 3-SN, 5-SN, and 6-SN mode is enabled, a configurable group of nine address bits in the range of address bits [21:8] and the configured top_addr_bits are essential in distributing the addresses between memory. Arm recommends that these bits are masked carefully to avoid memory aliasing. See 3.4.7.1 HN-F to SN-F memory striping in HN-F SAM on page 152
- The address bits masked in HN-F and RN SAM must be consistent. This requirement ensures that PrefetchTarget requests from an RN-F to SN-F are addresses by the same SN-F as SLC miss from the HN-F to SN-F.

3.4.7.4 HN-F SLC and SF flexibile addressing

CMN-700 HN-F supports flexible Set and Tag address for both SLC and SF. This allows for custom hashing or sliding of *Physical Address* (PA) bits to index into the SLC and SF cache. It is also useful when RNSAM's hierarchical hashing scheme is used because it can lead to unused sets in SLC and SF in certain use cases.

HN-F provides one 64-bit flex register each for SLC and SF which can be used for programming various modes and masks in custom implementations.

The default implementation of HN-F uses consecutive setaddr bits starting from bit 6 of the PA. For example, a system with 52 bits of PA and the SLC containing 1024 sets requires 10 bits to address into the cache. So the setaddr bits are PA[15:6]. The remaining address bits PA[51:16] are saved in the SLC RAM as "tag".

Generally, the contents of PA for the default set/tag addressing is:

{TAG, SETADDR, LINEOFFSET} <= PA[`PA_WIDTH_PARAM`-1:0]

Similarly, for SLC/SF victims, the PA is generated as an inverse of the previous equation (cache line aligned):

PA[`PA_WIDTH_PARAM`-1:0] <= {TAG, SETADDR, `6'b00`}

For information about modifying the Setaddr and Tag contents, see *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*.

The HN-F flexible Set and Tag address supports two modes of programming:

- Sliding Set Address mode:

In this mode, set address start bit is configurable to accommodate hierarchical hashing in RNSAM (cluster mask is set to 64bytes). Set address start bit is programmed based on the number of clusters enabled in the Hierarchal hashing based on below equation:
Set address start bit = 6 + log2 (number of clusters)

When this mode is enabled, the SLC/SF setaddr is shifted as shown below. All remaining bits are used as Tag bits stored in the SLC/SF cache.

PA[PA_WIDTH_PARAM-1:0] = {TAG1, SETADDR, TAG0, 6'b00}

TAG = {TAG1, TAG0}, TAG0 bits are based on number of clusters

- Programming registers: (For a 4 cluster example, Set address start bit = 8)
  - cmn_hns_pa2setaddr_sf.setaddr_startbit_sf = 4'b1000;
  - cmn_hns_pa2setaddr_slc.setaddr_startbit_slc = 4'b1000;
  - por_mxp_p#{index}_pa2setaddr_slc.setaddr_startbit_slc_p#{index} = 4'b1000;
  - por_mxp_p#{index}_pa2setaddr_sf.setaddr_startbit_sf_p#{index} = 4'b1000;

- Shuttering Set Address Mode

  Alternately HN-F flexible Set address bits are selected by shuttering precise bit locations in the PA. This controllable selection is required to enable RNSAM hierarchical cluster mask capability. The shuttering scheme must only have one group of repetitive bits shuttered.
  PA[PA_WIDTH_PARAM-1:0] = {TAG1, SETADDR1, TAG0 (shuttered bits), SETADDR0, 6'b00}

  SETADDR = {SETADDR1, SETADDR0}, SETADDR0 bits are based on cluster mask.

  TAG = {TAG1, TAG0}, TAG0 bits are based on number of clusters

  - Programming registers: (For a given system, Cluster interleaving = 4KB & Clusters# = 8)
    - cmn_hns_pa2setaddr_slc. setaddr_shutter_mode_en = 1'b1
    - cmn_hns_pa2setaddr_slc.setaddr_indx_[0-5]_shutter = 3'b000 //pass-through
    - cmn_hns_pa2setaddr_slc.setaddr_indx_[6-12]_shutter = 3'b011 // shift by 3 bits
    - cmn_hns_pa2setaddr_sf. setaddr_shutter_mode_en = 1'b1
    - cmn_hns_pa2setaddr_sf.setaddr_indx_[0-5]_shutter = 3'b000 //pass-through
    - cmn_hns_pa2setaddr_sf.setaddr_indx_[6-12]_shutter = 3'b011 // shift by 3 bits
    - repeat above programming in por_mxp#{index}_pa2setaddr_slc and por_mxp#{index}_pa2setaddr_sf registers as well

  
  Note

  1. Index ID in the programmable registers represents setaddr[12:0]
  2. If the Shutter bit is higher than the Set address index, then the shuttering mode must not be enabled
     Set_Addr_width = log2 ((CACHE_SIZE in MB)/ (Num_ways * 64))

So the shuttering must only be enabled if shutter_bit <= (set_addr_width +6)

3. In the setaddr_indx programming, once an index is programmed, all subsequent index must also have the same programming value. In the example 4KB interleaving, setaddr_indx_[12:6]_shutter must all be programmed to 3'b011.
4. These registers are present in both HN-F and MXP_port connected to the HN-F. The programming for setaddr must be the same in MXP and HN-F

Table 3-46: HN-F SAM parameters and configuration registers

HN-F SAM Feature	Applicable Parameters	Applicable configuration registers
Global	HNSAM_RCOMP_EN	hn_sam_region_cmp_addr_mask_reg
	HNSAM_RCOMP_LSB	
Default hashed	HNSAM_DEF_HASHED_GRP_EN	cmn_hns_sam_control cmn_hns_sam_6sn_nodeid cmn_hns_sam_sn_properties cmn_hns_sam_sn_properties1 cmn_hns_sam_sn_properties2
Non-hashed regions	HNSAM_NUM_NONHASH	cmn_hns_sam_memregion0-1 cmn_hns_sam_memregion0-1_end_addr cmn_hns_sam_nonhash_cfg1_memregion2-63 cmn_hns_sam_nonhash_cfg2_memregion2-63 cmn_hns_sam_sn_properties cmn_hns_sam_sn_properties1
Hashed regions	HNSAM_NUM_HTG	hn_sam_hash_addr_mask_reg cmn_hns_sam_htg_cfg1_memregion0-15 cmn_hns_sam_htg_cfg2_memregion0-15 cmn_hns_sam_htg_cfg3_memregion0-15 cmn_hns_sam_htg_sn_nodeid_reg0-15 cmn_hns_sam_htg_sn_attr0-15
Aggregated SA support	-	cmn_hns_sam_ccg_sa_nodeid_reg0-3 cmn_hns_sam_ccg_sa_attr0-3

HN-F SAM Feature	Applicable Parameters	Applicable configuration registers
CPAG support	-	cmn_hns_cml_port_aggr_grp0-31_add_mask cmn_hns_cml_port_aggr_grp_reg0-12 cmn_hns_cml_port_aggr_ctrl_reg cmn_hns_cml_port_aggr_ctrl_reg1-6
Custom hash	HNSAM_CUSTOM_REGS	hnf_generic_regs0-7


Note

- Parameters and registers in the same row are not necessarily related
- Ranges in register names, m-n, indicates the register repeats (m - n + 1) times

3.4.8 HN-I SAM

To simplify mapping and ordering of downstream endpoint address space, the HN-I SAM maps an incoming address to a target endpoint that is connected downstream behind HN-I.

The endpoint can be one of the following:

- Peripheral with memory-mapped I/O space, such as UART or GPIO
- Physical memory, such as SRAM or FLASH


Note

Arm recommends that the HN-I SAM is only programmed during the boot process.

To map and order the address space of these endpoints, the HN-I SAM supports:

- Up to three Address Regions

  
  Note

  Address Regions 1, 2, and 3 must not overlap.

- One Order Region of configurable size for each Address Region
- A default Address Region, Address Region 0

Each Address Region can be programmed as either peripheral or physical memory.


Note

By default, each Address Region is mapped to peripheral memory.

Physical

- Follows normal memory ordering guarantees.
- Order Region programming function output does not matter.

Peripheral

- Follows device memory ordering guarantees.
- These Address Regions can be further divided into smaller address spaces that are known as Order Regions. Device memory ordering guarantees are maintained within each Order Region.
- To enforce strict ordering for a specific Address Region, program its Order Region size to `6'b111111`.


Caution

If there is potential for new requests to fall into a newly configured Address Region or Order Region, and these requests require ordering with respect to the existing outstanding requests, the corresponding Address Region register must be disabled.


Note

- HN-I does not support write streaming from RN-Fs. HN-I sends CompDBID in response to RN-F write requests with ReqOrder and ExpCompAck. This response breaks the OWO as writes can be dispatched on AXI out of order.
- OWO is still supported from RN-I, RN-D, and CML HA nodes.

The minimum address granularity for Address Regions and Order Regions is 4KB. This size is equivalent to the minimum subordinate address space granularity in AXI/ACE-Lite. Therefore, the base address in the Address Region {1, 2, 3} Configuration Registers only includes bits[`REQ_ADDR_WIDTH`−1:12].


Caution

There are 128 unique AxIDs available for use when an address region is marked as physical memory. If HN-I is configured for more than 128 outsanding transactions to AXI (`NUM_RRT_REQS` + `NUM_AXI_REQS` > 128), physical memory will be treated as peripheral memory and all accesses will be sent with the same AxID.

Address Region 0

By default, the entire address space of a given HN-I is mapped to Address Region 0. All transactions to this region are kept in order.

The default Order Region size in Address Region 0 is `6'b111111`, which covers the entire HN-I address space. The Order Region size can also be configured to:

- `6'b101000` when `REQ_ADDR_WIDTH`==52
- `6'b100100` when `REQ_ADDR_WIDTH`==48
- `6'b100000` when `REQ_ADDR_WIDTH`==44

Address Region 0 is always valid. Therefore, the Address Region 0 Configuration Register does not define a Valid bit.

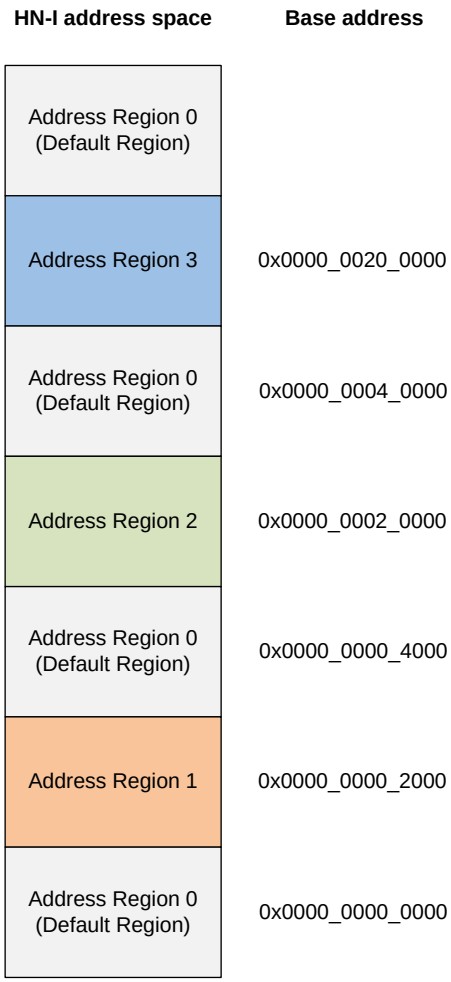
See 4.3.9.6 por_hni_sam_addrregion0_cfg on page 575.

3.4.8.1 HN-I SAM example configuration

This example system configuration for HN-I SAM uses three Address Regions and an Order Region within each Address Region.

The following figure shows the high-level configuration of the address space and the base addresses of each Address Region.

Figure 3-57: HN-I address space example



Note

In each Address Region Configuration Register, the following bitfields use the default value:

- ser_all_wr
- ser_devne_wr
- pos_early_wr_comp_en
- pos_early_rdack_en

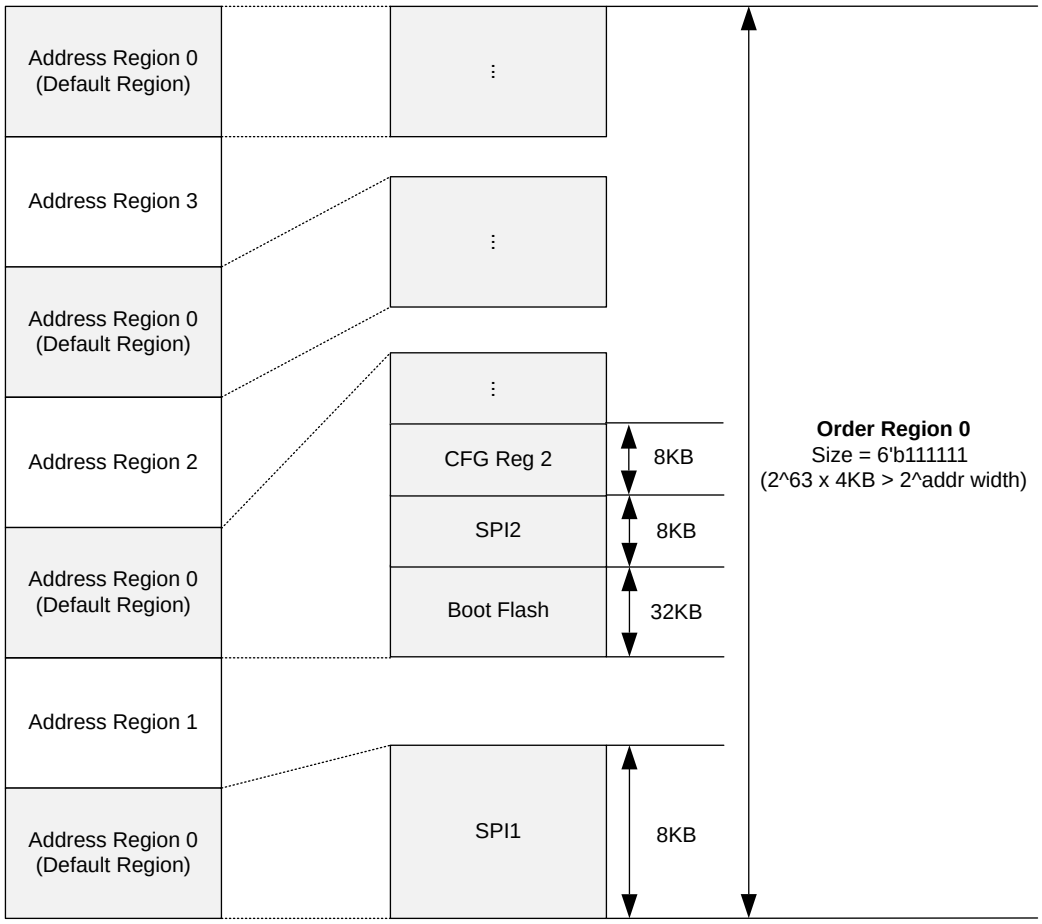
Each address region in the HN-I SAM is divided into one or more order regions. The width of all order regions within a single address region is the same. All accesses within an order region are kept in order, in other words sent with the same AxID value, unless the address region is marked as physical memory.

If the address region is marked as physical memory, only accesses to the same address are kept in order. In other words, accesses within an order region are given unique AxID values, unless they target the same address, where they are sent with the same AxID.

Address Region 0

The following figure shows the example configuration for Address Region 0.

Figure 3-58: Address Region 0 configuration



The size of the maximum peripheral address space in Address Region 0 is 32KB (Boot Flash). Requests targeting this Boot Flash must be kept in order. By configuring the Order Region 0 size to 32KB, requests with addresses from `0x0000_0000_0000` to `0x0000_0000_8000` are ordered. However, because Boot Flash is not aligned to the 32KB boundary (`0x0000_0000_4000` to `0x0000_0000_C000`), requests might be out of order and cause issues. To ensure all requests to Boot Flash are ordered, the Order Region 0 size must be configured to at least 64KB.

In this configuration, requests to SPI1, SPI2, and CFG Reg 2 are also ordered with respect to requests to Boot Flash. Instead, to optimize performance, Address Region 0 can have a SAM with Boot Flash aligned to the 32KB boundary and the Order Region 0 size can be configured to 32KB.

The following table shows the configured values for the Address Region 0 Configuration Register, por_hni_sam_addrregion0_cfg.

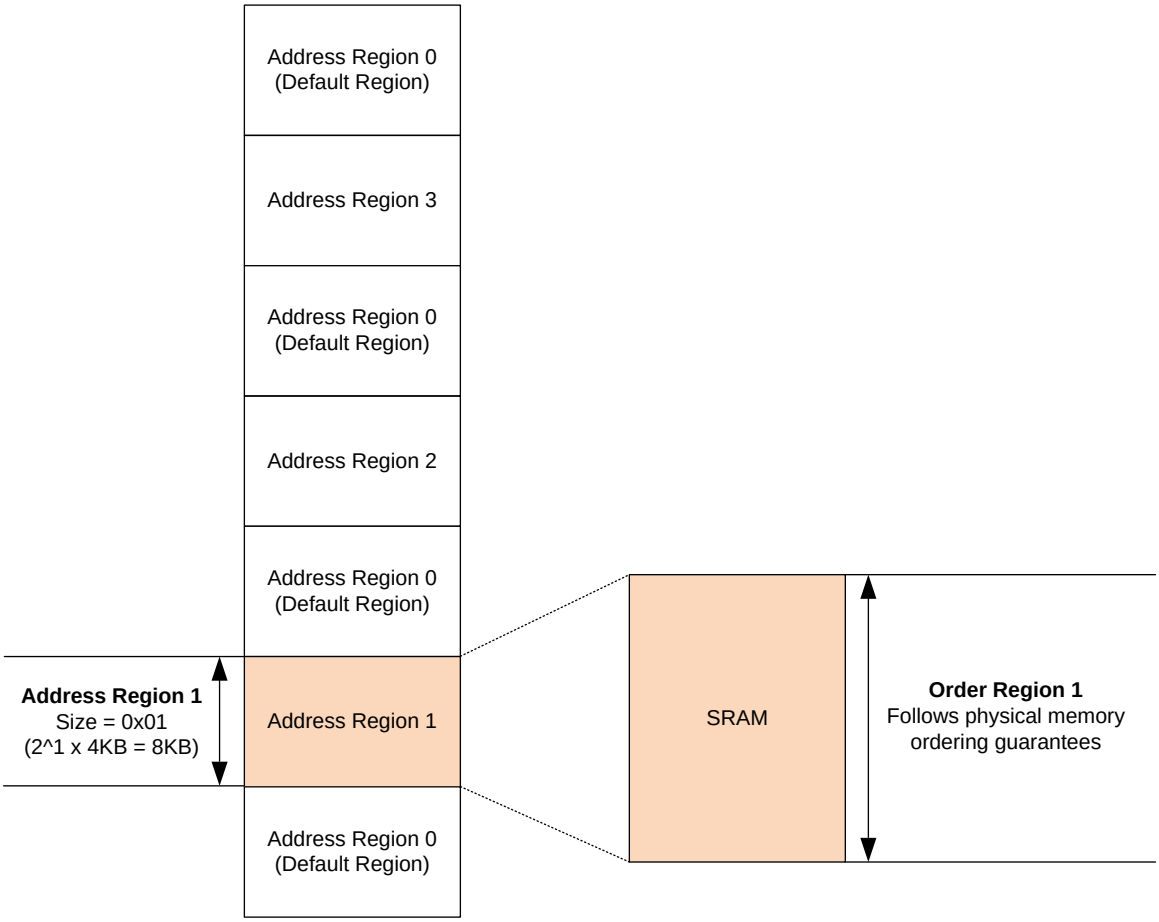
Table 3-47: Address Region 0 Configuration Register

Bits	Field name	Configured value
[5:0]	order_reg_size	`6'h4`
[58]	physical_mem_en	`1'b0`
[59]	ser_all_wr	`1'b0`
[60]	ser_devne_wr	`1'b0`
[61]	pos_early_rdack_en	`1'b1`
[62]	pos_early_wr_comp_en	`1'b1`

Address Region 1

The following figure shows the example configuration for Address Region 1.

Figure 3-59: Address Region 1 configuration



Address Region 1 starts at base address `0x0000_0000_2000` and is 8KB in size. Because there is SRAM behind this region, it is mapped as physical memory. The entire Address Region 1 is considered as one Order Region. Therefore, ordering is maintained between all requests to the overlapping cache line region (64B).

The following table shows the configured values for the Address Region 1 Configuration Register, por_hni_sam_addrregion1_cfg.

Table 3-48: Address Region 1 configuration

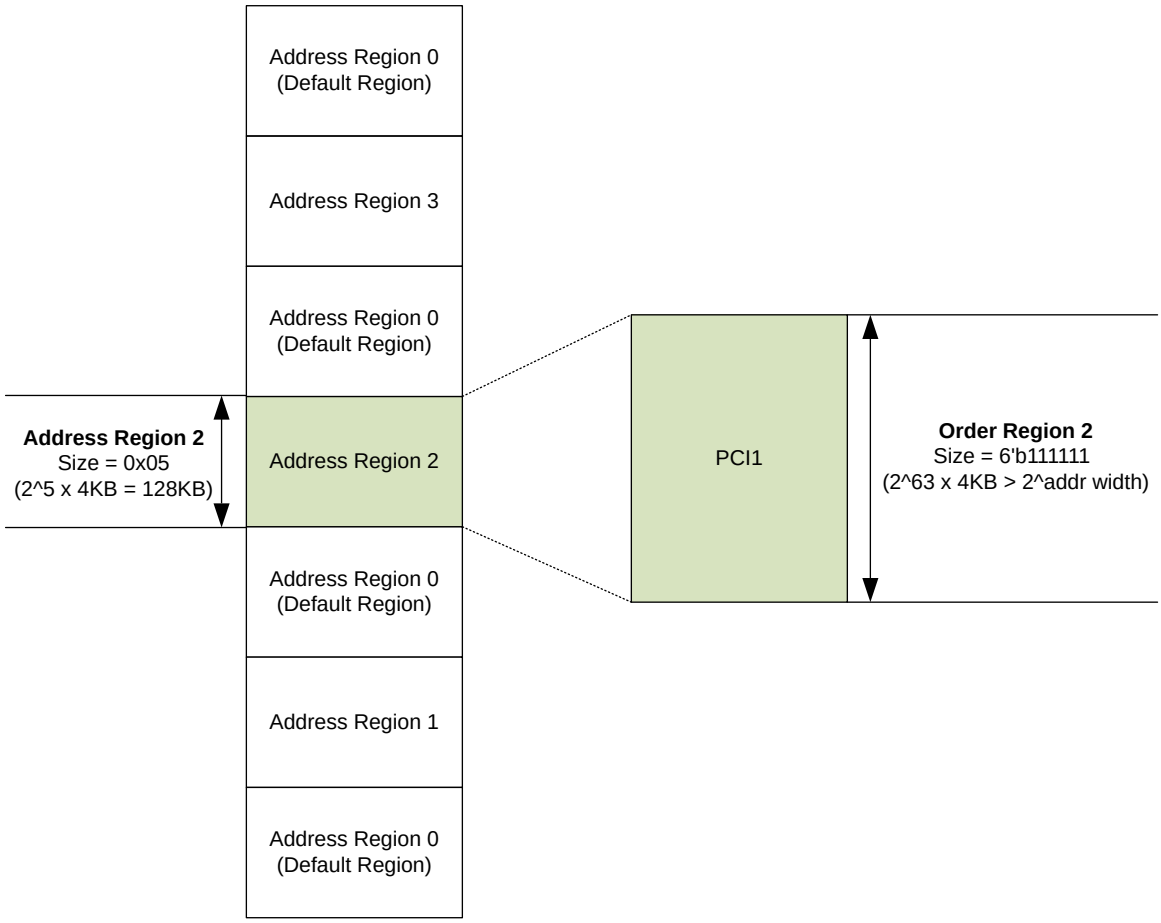
Bits	Field name	Configured value
[5:0]	order_reg_size	`6'h1`
[15:10]	addr_region_size	`6'h1`
[55:16]	base_addr	`40'h0000_0000_2`
[58]	physical_mem_en	`1'b1`
[59]	ser_all_wr	`1'b0`

Bits	Field name	Configured value
[60]	ser_devne_wr	1'b0
[61]	pos_early_rdack_en	1'b1
[62]	pos_early_wr_comp_en	1'b1
[63]	valid	1'b1

Address Region 2

The following figure shows the example configuration for Address Region 2.

Figure 3-60: Address Region 2 configuration



Address Region 2 starts at base address `0x0000_0002_0000` and is 128KB in size. The Order Region 2 size (2^{63} × 4KB) is configured to the maximum value (`6'b111111`), so Address Region 2 is considered as one Order Region. PCI1 occupies the entire Order Region, so all PCI1 requests are ordered.

The following table shows the configured values for the Address Region 2 Configuration Register, por_hni_sam_addrregion2_cfg.

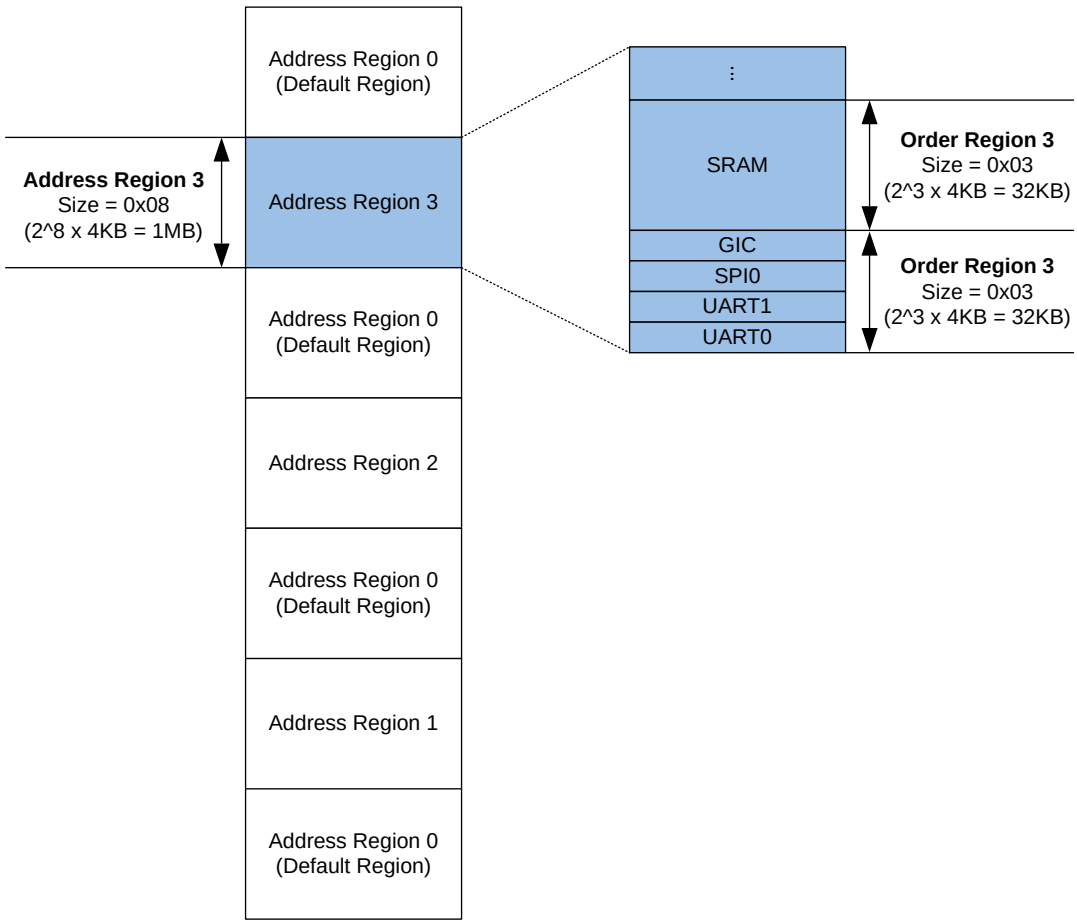
Table 3-49: Address Region 2 configuration

Bits	Field name	Configured value
[5:0]	order_reg_size	6'b111111
[15:10]	addr_region_size	6'h5
[55:16]	base_addr	40'h0000_0002_0
[58]	physical_mem_en	1'b0
[59]	ser_all_wr	1'b0
[60]	ser_devne_wr	1'b0
[61]	pos_early_rdack_en	1'b1
[62]	pos_early_wr_comp_en	1'b1
[63]	valid	1'b1

Address Region 3

The following figure shows the example configuration for Address Region 3.

Figure 3-61: Address Region 3 configuration



Address Region 3 starts at base address `0x0000_0020_0000` and is 1MB in size. The Order Region 3 size of 32KB is less than the Address Region 3 size of 1MB, resulting in a total of 32 Order Regions. GIC, SPI0, UART1, and UART0 map to one Order Region, therefore all requests to these peripherals are ordered. SRAM also maps to one Order Region, therefore all requests to SRAM are ordered. Because SRAM maps to a separate Order Region from GIC, SPI0, UART1, and UART0, requests to SRAM and requests to GIC, SPI0, UART1, and UART0 are not ordered.

The following table shows the configured values for the Address Region 3 Configuration Register, por_hni_sam_addrregion3_cfg.

Table 3-50: Address Region 3 configuration

Bits	Field name	Configured value
[5:0]	order_reg_size	`6'h3`
[15:10]	addr_region_size	`6'h8`
[55:16]	base_addr	`40'h0000_0020_0`

Bits	Field name	Configured value
[58]	physical_mem_en	1'b0
[59]	ser_all_wr	1'b0
[60]	ser_devne_wr	1'b0
[61]	pos_early_rdack_en	1'b1
[62]	pos_early_wr_comp_en	1'b1
[63]	valid	1'b1


Note

In HN-P, SAM programming does not apply to requests from PCIe RN-Is or PCIe RN-Ds. PCIe RN-Is and PCIe RN-Ds are designated by the pcie_mstr_present configuration bit in the RN-I or RN-D node. Requests from these node types always assume that traffic is directed to endpoint memory space. The processing of requests from these sources is optimized according to PCIe ordering rules.

3.4.9 SAM memory region size configuration

GIC, hashed, and non-hashed memory regions support various sizes. Each memory partition must be individually programmed in the SAM registers.

RN SAM and HN-F SAM support the following memory partition sizes:

GIC

64KB, 128KB, 256KB, and 512KB.

Hashed and non-hashed

64MB for RN SAM and HN-F SAM up to maximum addressable space (2^{PA_WIDTH}).

The following table lists the GIC memory partition size encodings that are used to program the RN SAM and HN-F SAM registers.

Table 3-51: RN SAM and HN-F SAM configuration register GIC memory partition sizes

Memory partition size	regionX_size value
GIC	
64KB	3'b000
128KB	3'b001
256KB	3'b010
512KB	3'b011

The following table lists the hashed and non-hashed memory partition size encodings that are used to program the RN SAM and HN-F SAM registers.

Table 3-52: RN SAM and HN-F SAM configuration register hashed and non-hased memory partition sizes

Memory partition size	regionX_size value
Hashed and non-hashed	
64MB	7'b0000000
128MB	7'b0000001
256MB	7'b0000010
512MB	7'b0000011
1GB	7'b0000100
2GB	7'b0000101
4GB	7'b0000110
8GB	7'b0000111
16GB	7'b0001000
32GB	7'b0001001
64GB	7'b0001010
128GB	7'b0001011
256GB	7'b0001100
512GB	7'b0001101
1TB	7'b0001110
2TB	7'b0001111
4TB	7'b0010000
8TB	7'b0010001
16TB	7'b0010010
32TB	7'b0010011
64TB	7'b0010100
128TB	7'b0010101
256TB	7'b0010110
512TB	7'b0010111
1PB	7'b0011000
2PB	7'b0011001
4PB	7'b0011010

The RN SAM also outputs the target type of the device along with the target ID for the RN to use in various optimizations. The following table lists the target type encodings.

Table 3-53: Device target types

Device type	Target type
HN-F	3'b000
HN-I	3'b001
CCG RA	3'b010
HN-Ps	3'b011
PCI-CCG RA	3'b100

The following table contains RA SAM configuration register memory partition sizes and encodings.

Table 3-54: RA SAM configuration register memory partition sizes

Memory partition size	regionX_size value
64KB	6'b000000
128KB	6'b000001
256KB	6'b000010
512KB	6'b000011
1MB	6'b000100
2MB	6'b000101
4MB	6'b000110
8MB	6'b000111
16MB	6'b001000
32MB	6'b001001
64MB	6'b001010
128MB	6'b001011
256MB	6'b001100
512MB	6'b001101
1GB	6'b001110
2GB	6'b001111
4GB	6'b010000
8GB	6'b010001
16GB	6'b010010
32GB	6'b010011
64GB	6'b010100
128GB	6'b010101
256GB	6'b010110
512GB	6'b010111
1TB	6'b011000
2TB	6'b011001
4TB	6'b011010
8TB	6'b011011
16TB	6'b011100
32TB	6'b011101
64TB	6'b011110
128TB	6'b011111
256TB	6'b100000
512TB	6'b100001
1PB	6'b100010
2PB	6'b100011
4PB	6'b100100

3.4.10 GIC communication over AXI4-Stream ports

CMN-700 supports optional manager/subordinate *AXI4-Stream* (A4S) ports on RN-I, RN-D, SN-F, and MXP RN-F ports for communication between a *Generic Interrupt Controller* (GIC) and CPUs. Certain requirements apply to the A4S routing and signaling.

CMN-700 also supports transmission of GIC information across CCIX links for CML SMP configurations.

More system-level information is available in the *Arm® Neoverse™ N1 hyperscale reference design GIC-600 Integration using CMN-600 AXI4-Stream Interfaces White Paper* on request.

A4S routing

The A4S ports are addressed according to Logical ID, which are assigned sequentially from 0 to the number of A4S ports. To send a packet from one A4S port to the destination A4S port, assign the TDEST to one of the following Logical IDs:

- The Logical ID of the target A4S port
- The Logical ID of the CML gateway for GIC traffic targeting the other chip

The discovery process returns the number of A4S ports and Logical ID information for each A4S port. This information is collected by reading corresponding RN-I, RN-D, and XP unit information registers. For more information about the discovery process, see 3.5 Discovery on page 193.

Other requirements

- The PUB_DESTID associated with the GICD A4S port must drive the CML gateway GICD_DESTID input strap. The PUB_DESTID value is included in the CMN ID-mapping file that is created during the IP rendering process. See the *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.
  GICD drives the CMN-700 RXA4STRI[5:0] input (6 MSB bits of GICD ICDRTDEST), indicating the CCIX link of the target chip. GICD also drives the RXA4STDEST[9:0] (10 LSB bits of GICD ICDRTDEST) of CML gateway for CML SMP configurations.
- The A4S manager must assert valid irrespective of ready state to transmit data.

3.4.11 Default XY routing behavior

By default, CMN-700 uses an XY routing algorithm to decide which direction to route flits within the mesh. At each MXP, the XID and YID values of the target MXP and the current MXP are compared to determine the routing direction.

Routing directions are referred to by the mesh port that the MXP routes the flit through. For example, if the MXP routes the flit northwards, then the flit is sent through the north mesh port.

If there is a mismatch between the target MXP XID and the current MXP XID, then the MXP uses the following rule to decide the routing direction:

- If target MXP XID > current MXP XID, then route eastwards
- Otherwise, route westwards

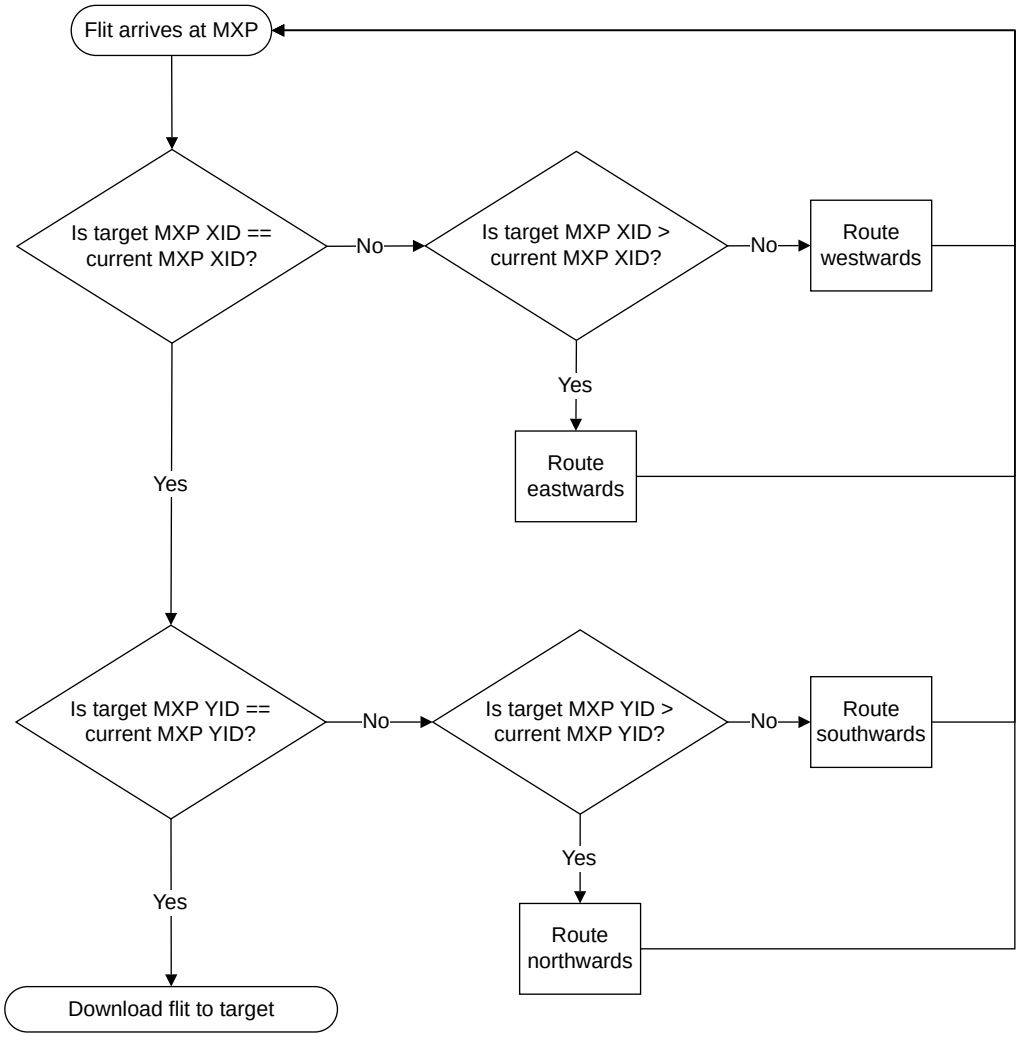
If the target MXP XID and the current MXP XID match, then the flit routing components are compared against the YID of the MXP. If YIDs do not match, then the MXP uses the following rule to decide the routing direction:

- If target MXP YID > current MXP YID, then route northwards
- Otherwise, route southwards

If the target MXP XID and YID match the current MXP XID and YID, then the flit has reached the target MXP. At this point, the flit is downloaded to the target device.

The following figure shows the default XY routing flow.

Figure 3-62: Default XY routing flow



You can configure CMN-700 to override the default XY routing pattern for specific source-target pairs in the mesh. For more information about this feature, see 3.4.12 Non-XY routing on page 178.

3.4.12 Non-XY routing

You can configure up to 16 source-target pairs in the CMN-700 mesh to route CHI traffic against the default XY routing algorithm. Non-XY routing can improve the efficiency of traffic flow by reducing hotspots in the mesh layout.

By default, CMN-700 uses an XY routing mechanism to route flits through the mesh. For more information about the default XY routing mechanism, see 3.4.11 Default XY routing behavior on page 176.

You can configure any source-target pair of XPs in your mesh configuration to use non-XY routing, up to a maximum of 16 pairs.

Enable this optional feature using the `XY_OVERRIDE_CNT` parameter, which supports values 0, 2, 4, 8, or 16. The value represents the number of source-target pairs which use non-XY routing. To define the non-XY routing XP pairs and their behavior, you can program the por_mxp_xy_override_sel_* registers at boot.

This feature only applies to the CHI channels, REQ, RSP, DAT, and SNP, not to the PUB channel.

3.4.12.1 Configuring non-XY routing behavior

A boot-programmable static *Lookup Table* (LUT) in each XP controls non-XY routing.

Enable support for this feature by setting the `XY_OVERRIDE_CNT` parameter to a non-zero value. See 3.4.12 Non-XY routing on page 178 and 2.5.2 Mesh sizing and top-level configuration on page 25.

Eight 64-bit boot-programmable registers control the non-XY routing feature, por_mxp_xy_override_sel_* registers. These registers support override of the route paths for up to 16 source-target XP pairs.

The contents of the por_mxp_xy_override_sel_* registers represent a static LUT.

The following table shows the format of each entry in the LUT.

Table 3-55: LUT entry format

Field	Description
<SRCID>	The source ID of the source-target pair that is enabled for XY override
<TGTID>	The target ID of the source-target pair that is enabled for XY override
CAL TGT PRESENT	CAL TGT Presence Indication for XY Route Override of all devices behind CAL
YX turn enable	Allows YX turn in the XP
XY route override	Enables flit XY route override in the XP

The following table shows the structure of a single non-XY routing register.

Table 3-56: por_mxp_xy_override_sel_* structure

Bitfield	Name
[63]	VALID
[62:59]	Reserved
[58:48]	srcid_1
[47]	Reserved
[46:36]	tgtid_1
[35]	Reserved
[34]	cal_tgt_present_1
[33]	yx_turn_enable_1
[32]	xy_override_enable_1
[31:27]	Reserved
[26:16]	srcid_0
[15]	Reserved
[14:4]	tgtid_0
[3]	Reserved
[2]	cal_tgt_present_0
[1]	yx_turn_enable_0
[0]	xy_override_enable_0

When routing flits between XPs, the XP compares the <SRCID> and <TGTID> flit fields against the entries in this LUT. This comparison, along with the YX turn enable and XY route override values for each XP, identify the route for the flit to take.

For the specific programming sequence to set up the LUT, see 4.4.5 Program non-XY routing registers on page 1132.

3.4.12.2 Rules for avoiding deadlocks in non-XY routing

You must follow various rules to ensure that the non-XY routing implementation is free of deadlocks.

In the default XY routing scheme, the following turns are forbidden:

- S → E
- N → W
- S → W
- N → E

For non-XY routing, these turns are allowed, but you must apply the following rules to avoid deadlocks. xj or xi represents the XID value of an XP, and yj or yi represents the YID value of an XP.

- If N → W turn is allowed at $XP_{xi,yi}$, then S → E turn is disallowed at every $XP_{xj,yj}$ where (xj < xi) and (yj < yi).

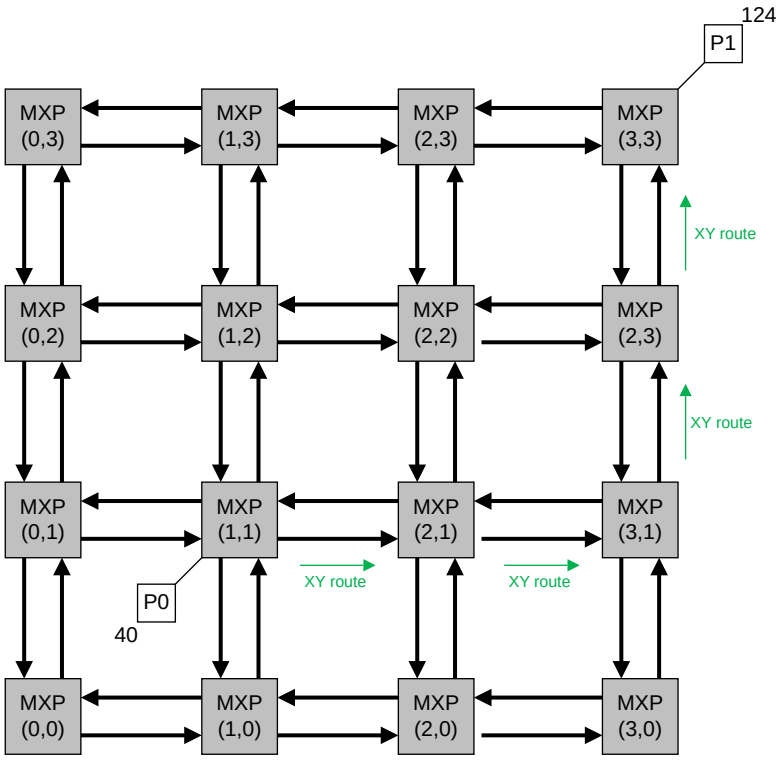
- If S → E turn is allowed at $XP_{xi,yi}$, then N → W turn is disallowed at every $XP_{xj,yj}$ where (xj > xi) and (yj > yi).
- If N → E turn is allowed at $XP_{xi,yi}$, then S → W turn is disallowed at every $XP_{xj,yj}$ where (xj > xi) and (yj < yi).
- If S → W turn is allowed at $XP_{xi,yi}$, then N → E turn is disallowed at every $XP_{xj,yj}$ where (xj < xi) and (yj > yi).

3.4.12.3 Non-XY routing examples

As an example, consider a flit that is uploaded from decimal source NodeID 40 and targets decimal NodeID 124 on a 4 × 4 mesh configuration.

The following figure shows the default routing of the flit without non-XY routing.

Figure 3-63: Default XY routing example



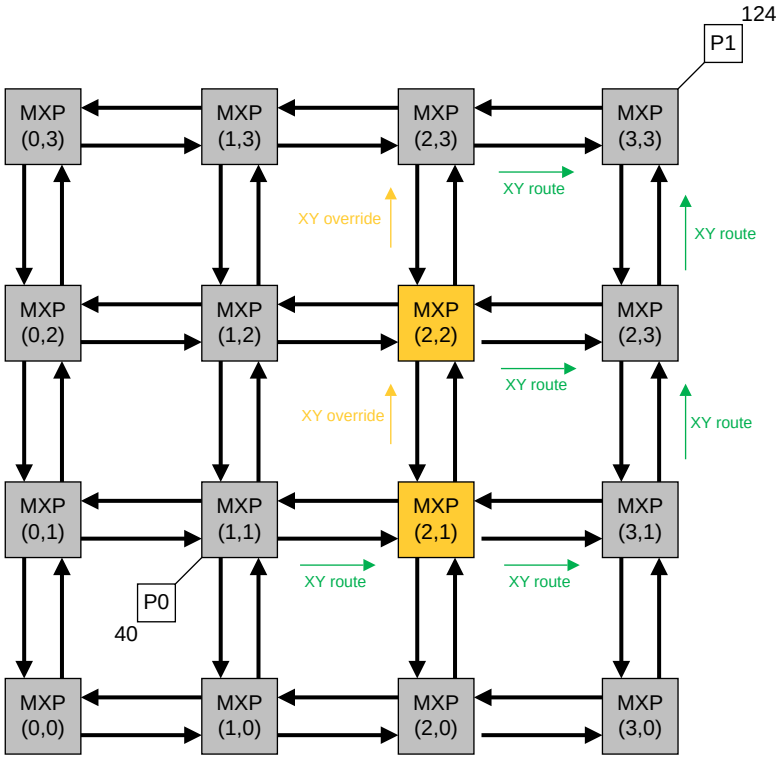
According to the standard XY routing algorithm, the flit follows the following route:

1. MXPs route the flit in the east direction until the flit reaches $MXP_{(3,1)}$
2. MXPs route the flit in the north direction to $MXP_{(3,3)}$, where the target node downloads the flit

XY override enabled, YX turn disabled

The following figure shows the default routing of a flit and the XY override route.

Figure 3-64: XY override enabled and YX turn disabled routing example



In the example, the non-XY routing registers in $MXP_{(2,1)}$ and $MXP_{(2,2)}$ are configured to override the XY route for a set of source-target pairs. This set includes NodeID40 and NodeID124, so the flit follows the following route:

1. $MXP_{(1,1)}$ routes the flit in the east direction
2. $MXP_{(2,1)}$ and $MXP_{(2,2)}$ route the flit in the north direction, since their configuration has XY override enabled
3. There is no override set in $MXP_{(2,3)}$. Therefore, the MXP routes the flit in the east direction according to the default XY routing algorithm.
4. At $MXP_{(3,3)}$, the flit has reached its destination, and the target node downloads the flit

If the XY override option is enabled and YX turn option is disabled, the following assumptions and constraints apply to the routing algorithm:

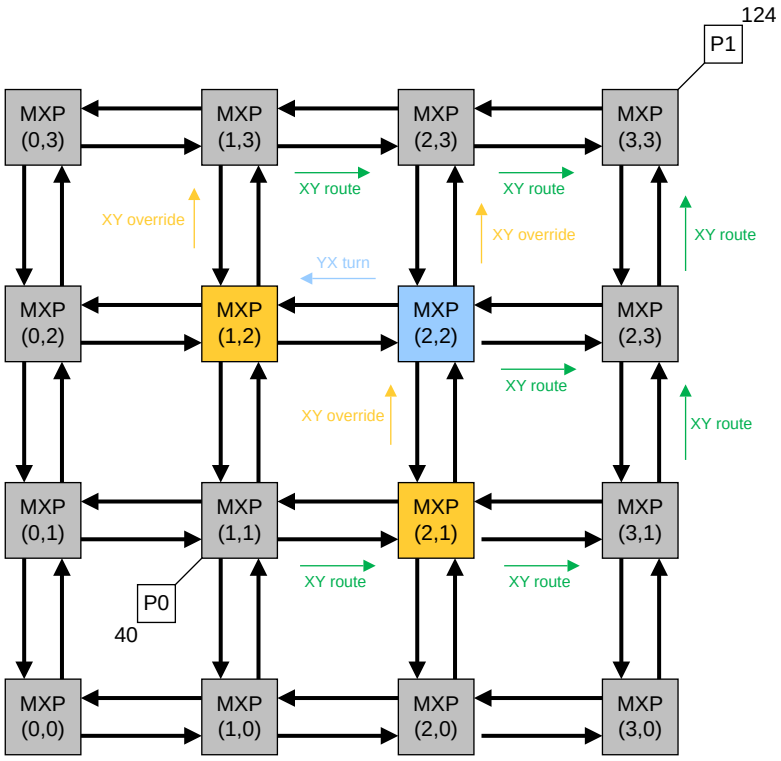
- If target MXP YID ≥ current MXP YID and a northern mesh port is present, then route northwards

- If target MXP YID < current MXP YID and a southern mesh port is present, then route southwards
- If target MXP YID == current MXP YID, a southern mesh port is present, and a northern mesh port is absent, then route southwards
- Otherwise follow the default XY routing algorithm

Both XY override and YX turn enabled

The following figure shows the default routing of a flit and an example XY override and YX turn enabled route.

Figure 3-65: XY override and YX turn enabled routing example



In the example, to enable non-XY routing, the following configurations have been made for a set of source-target pairs, including NodeID40 and NodeID124:

- Non-XY routing registers in $MXP_{(1,2)}$ and $MXP_{(2,1)}$ are configured to override the XY route
- Non-XY routing registers in $MXP_{(2,2)}$ are configured to override the XY route and enable YX turns

Therefore, the flit follows the following route:

1. $MXP_{(1,1)}$ routes the flit in the east direction

2. $MXP_{(2,1)}$ routes the flit in the north direction, since its configuration has XY override enabled
3. $MXP_{(2,2)}$ routes the flit in the west direction, since its configuration has XY override and YX turn enabled
4. $MXP_{(1,2)}$ routes the flit in the north direction, since its configuration has XY override enabled
5. There is no override set in $MXP_{(1,3)}$ and $MXP_{(2,3)}$. Therefore, the flit is routed in the east direction according to the default XY routing algorithm.
6. At $MXP_{(3,3)}$, the flit has reached its destination, and the target node downloads the flit

If XY route override and YX turn are enabled, the following assumptions and constraints apply to the routing algorithm:

- If target MXP XID ≤ current MXP XID and an eastern mesh port is present, then route eastwards
- If target MXP XID > current MXP XID and a western mesh port is present, then route westwards
- If target MXP XID == current MXP XID, a western mesh port is present, and an eastern mesh port is absent, then route westwards
- Otherwise, follow the default XY routing algorithm

3.4.13 Cross chip routing and ID mapping

IDs are generated and used to route CML_SMP protocol messages across multiple chips.

The following acronyms are used in this section:

- *Request Agent ID* (RAID)
- *Home Agent ID* (HAID)
- *Logical Device ID* (LDID)

By default, LDIDs are uniquely assigned within a device type. For example, RN-Fs in the system can be assigned LDIDs 0-n. RN-Is can be assigned LDIDs 0-m. RN-Ds can be assigned LDIDs 0-k.

The following rules apply to CML configurations with HN-F SF clustered mode disabled:

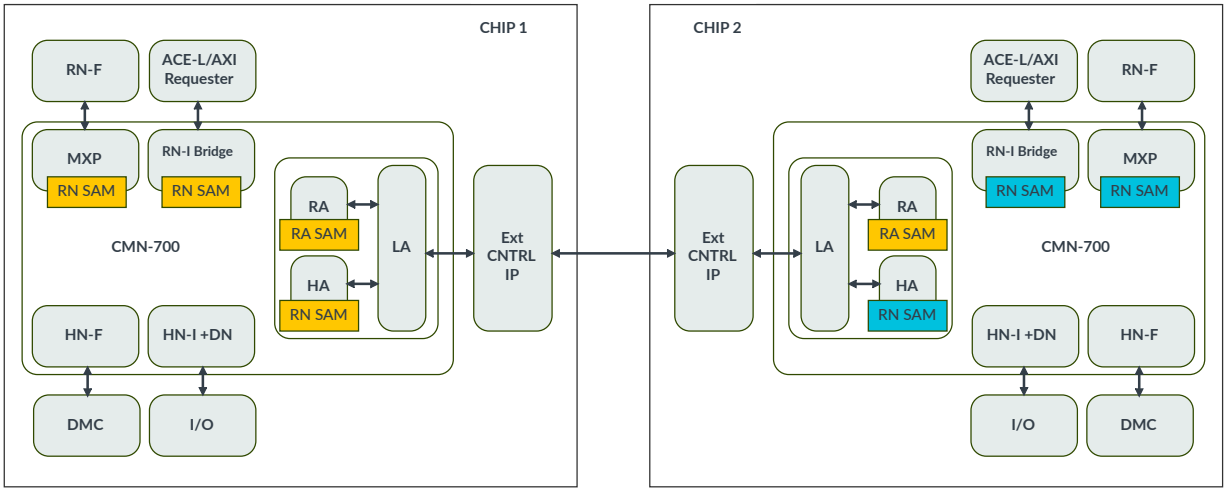
- RAID usage:
  - Confined to CML gateway devices only.
  - All local and remote components that are visible to CMN-700 use and operate on sequentially assigned LDIDs. CCG devices bidirectionally map each RAID to an LDID.
- RN-F default LDID assignment:
  - Local RN-Fs are assigned LDIDs from 0-n, sequentially
  - Remote RN-Fs must be assigned LDIDs n+1 and above by the discovery software

If HN-F SF clustered mode is enabled and LDID override is used, the preceding rules do not apply. In this mode, local and remote RN-Fs can be assigned LDIDs from the full ID space to meet the

clustering requirement. The number of bits required for RN-F LDID values, the LDID_WIDTH, is determined by the total number of local and remote RN-F/RA nodes and determines the full ID space.. To maximize SF efficiency, assigned LDIDs must be sequential and the LDID space must not have any holes.

The following figure shows a basic multi-chip block diagram.

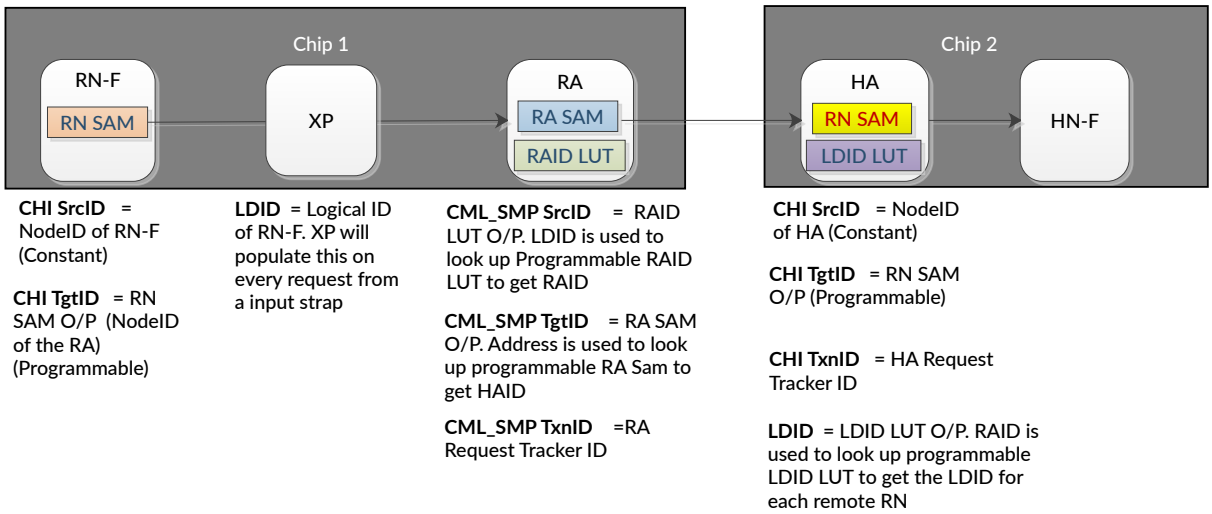
Figure 3-66: Multi-chip block diagram



Request from an RN-F to a remote HN-F

The following figure shows all IDs generated and used to route a request from a local RN-F on chip 1 to a remote HN-F on chip 2.

Figure 3-67: RN-F to remote HN-F IDs

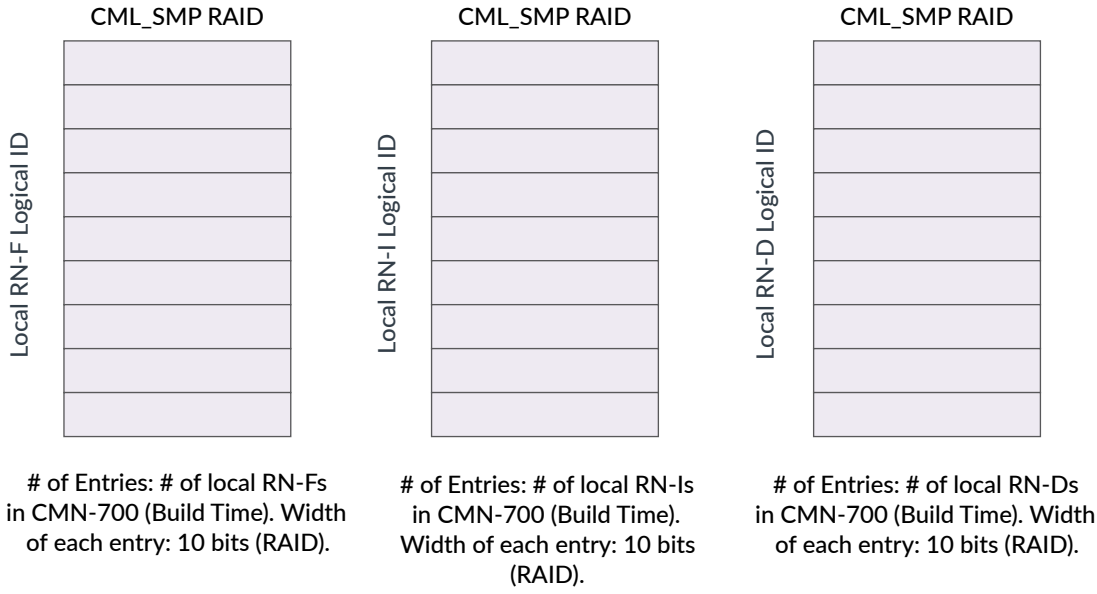


The flow for this process is as follows:

- The RN-F looks up the programmable RN SAM to populate the CHI target ID on a request.
- The XP populates the RN-F LDID on this request.
- When HN-F SF clustering is disabled, the LDIDs for local RN-Fs must not be changed. The build-time LDID assignments are discovered by reading any one of the HN-F cmn_hns_rn_cluster<X>_physid<Y> registers. A few cycles after reset, these registers are prepopulated with the LDIDs for local RN-Fs within that chip.
- If HN-F SF clustered mode is enabled:
  - The local RN-F LDID in por_mxp_p[0-5]_ldid_override register can be programmed to match the clustering requirements at each RN-F port. In clustered mode, the local RN-F LDIDs are not pre-programmed in the physical ID registers of the HN-F out of reset. Therefore, you must explicitly program these registers to suit the clustering requirements.
  - When HN-F SF clustered mode is enabled, CMN-700 allows override of the LDID of each RN-F. This LDID must be programmed in the 4.3.3.16 por_ccg_ra_rnf_ld0-127d_to_ovrd_ld0-127d_reg0-127ndex on page 347 registers of the CML-RA. Also, each override value must match the por_mxp_p[0-5]_ldid_override register value of the corresponding RN-F.
  - When HN-F SF clustered mode is enabled, CMN-700 allows override of the LDID of each RN-F. This LDID must be programmed in the registers of the CML-RA. Also, each override value must match the por_mxp_p[0-5]_ldid_override register value of the corresponding RN-F.
  - If HN-F SF clustered mode is enabled, there must be at least two cluster groups. In other words, you cannot cluster all RN-Fs into a single cluster group.
  - If HN-F SF clustered mode is enabled, and there is a single RN-F in the cluster, then it must always use the lowest device ID in that cluster. Consider a four-way clustering, where the device ID fields are `0b00`, `0b01`, `0b10`, and `0b11`. In this configuration, you cannot use device ID values of `0b01`, `0b10`, or `0b11` unless `0b00` is also in use.
- por_mxp_device_port_connect_ldid_info_p[0-5] captures the default LDID values assigned to the RN-Fs that are connected to the respective device port.
- The cmn_hns_rn_cluster_*_physid_reg* registers also contain fields to program the source type for each RN-F in the system. For all local RN-Fs, the source type must be programmed to the appropriate CHI protocol issue that the RN-F supports. For all remote RN-Fs, the source type must be programmed to `0b1100` (CHI-E) as CML-HA is a proxy for all remote RN-Fs.
- RN-Is and RN-Ds are internal to CMN-700 and get their logical ID assigned during CMN-700 generation. The RN-I or RN-D sends this LDID on every request.
- CML-RA contains programmable lookup tables, RAID LUTs, for each class of local RN (RN-F, RN-I, RN-D, and CML-HA). CML_SMP discovery software discovers all local RN-Fs, RN-Ds, and RN-Is, and programs their corresponding RAIDs in these LUTs.
- The LDID of the incoming request is used to look up these RAID LUTs and determine the CML SMP RAID. CML-RA also has CML RA SAM. This CML RA SAM is used to generate the HAID. This HAID is used as the target ID to route the CML request message.
- The build-time LDID assignment can be discovered by reading any of the por_ccg_ra_rnf_ldid_to_nodeid_reg registers.

The following figure shows the programmable registers during CML discovery.

Figure 3-68: Programmable registers during CML Discovery



The CML-HA contains a programmable register to program the local LDID for each remote CML RAID that communicates with local CHI HNs on a given chip or socket. Each entry in the programmable register also contains an RN-F bit to identify whether the remote CML-RA is a caching agent (RN-F) or not. If RN-I and RN-F have the same RAID, then you must only fill the RN-F details in the entry.

HN-Fs on the local chip use this LDID to track a line in its SF. Therefore a unique LDID assignment is required for each remote requesting caching agent. These unique LDIDs must not overlap with LDIDs assigned to local RN-Fs. It is assumed that these IDs are assigned after CML discovery is complete. For example, all the CML-RAs are discovered and assigned an RAID.

The following figure shows the programmable register for RAID to LDID during CML discovery.

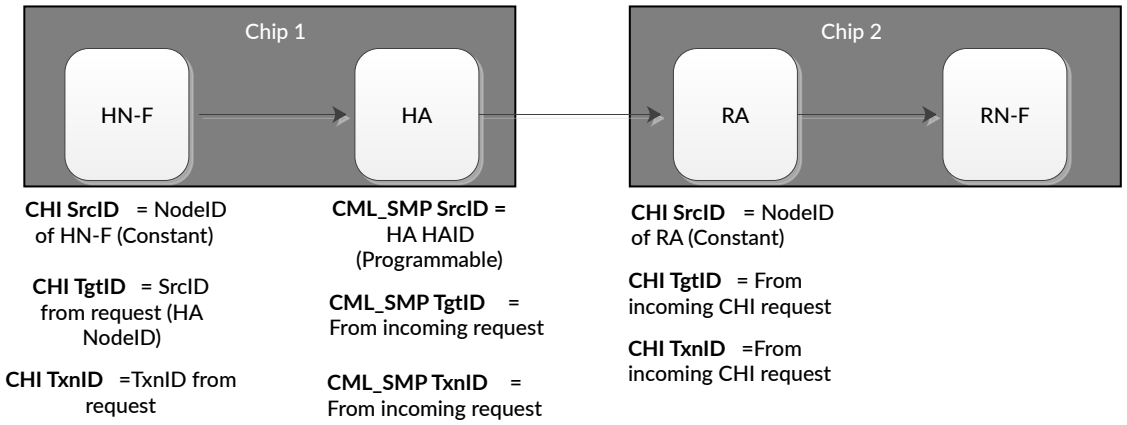
With the LDID passed to HN-F in all CHI REQ flits, HN-F uses this LDID as the true logical ID for SF tracking purposes. HN-F uses a logical ID vector in the SF. Depending on whether SF clustered mode is enabled or not:

- LDIDs can be uniquely tracked in the SF
- Multiple LDIDs can be aliased to a single logical ID

The total number of bits in the SF vector is calculated based on configuration parameters. You can also make the vector larger using the `NUM_ADD_SF_VECTOR` configuration parameter.

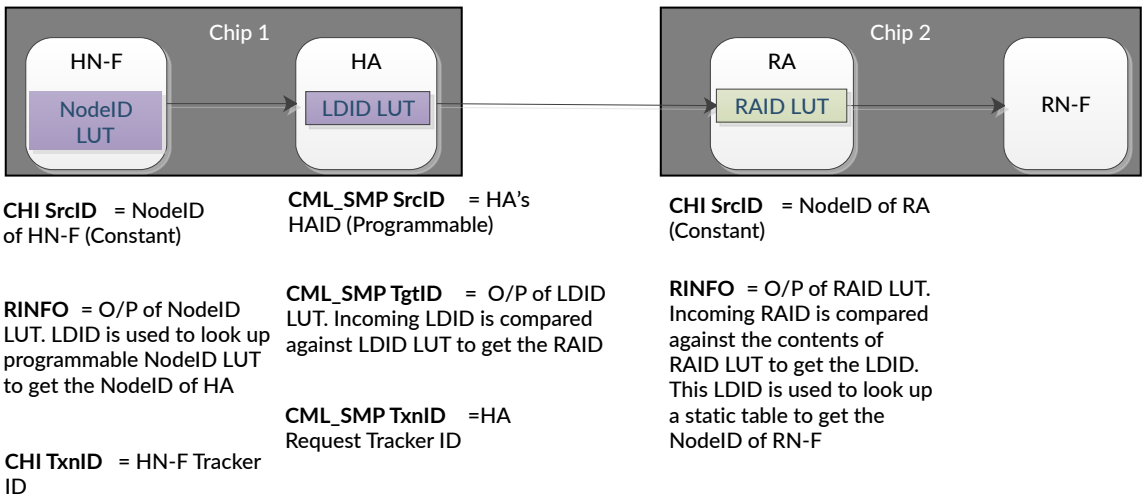
The following figure shows all generated IDs used to route a response from a remote HN-F on chip 2 to an RN-F on chip 1.

Figure 3-69: Remote HN-F to RN-F IDs



The following figure shows the flow of a snoop from an HN-F to a remote RN-F.

Figure 3-70: Snoop from HN-F to remote RN-F



The HN-F contains the following programmable LUT to program the CML-HA node ID of each remote caching agent. HN-F uses the unique LDID from the snoop vector to look up the physical CHI node ID of the CML-HA where snoops are sent to. The following table shows an example programming where logical IDs 0-7 are assigned to the local RN-Fs. The logical IDs 8-15 are assigned to the remote RN-Fs. Software assigns these IDs during the discovery process.

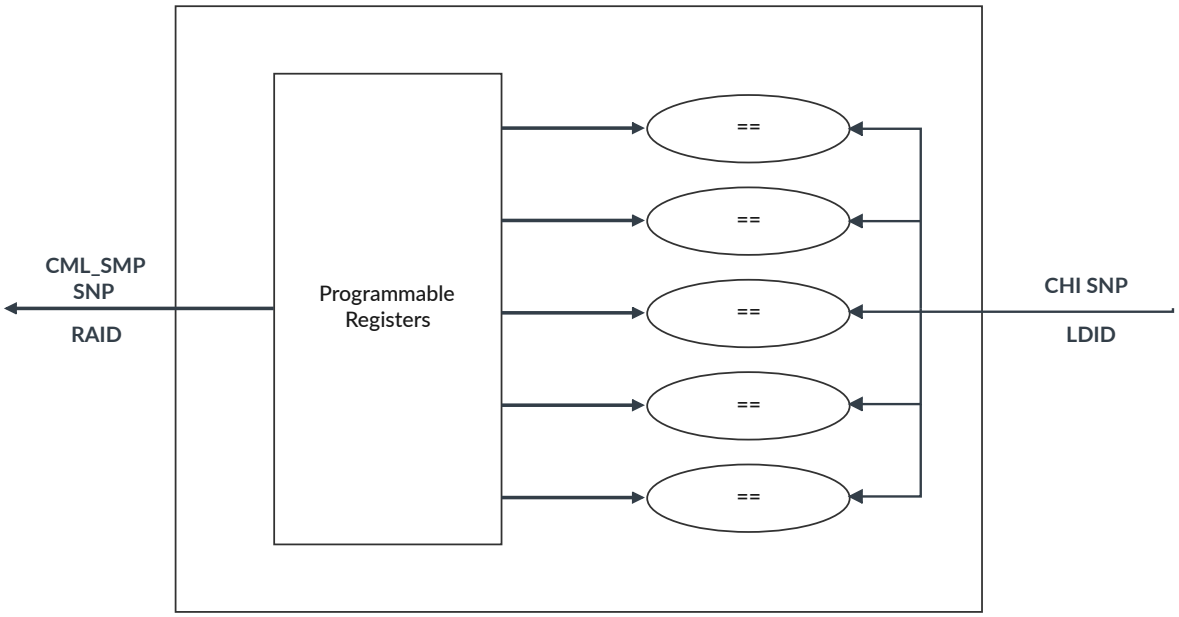
Table 3-57: Example program

Logical ID as index	HN-F programmable register	
	CHI node ID	ID valid
0	Local RN-F 0	1

Logical ID as index	HN-F programmable register	
	CHI node ID	ID valid
1	Local RN-F 1	1
2	Local RN-F 2	1
...	...	...
7	Local RN-F 7	1
8	CML-HA	1
9	CML-HA	1
...	...	...
15	CML-HA	1
16	Not programmed	0
...	Not programmed	0
n	Not programmed	0

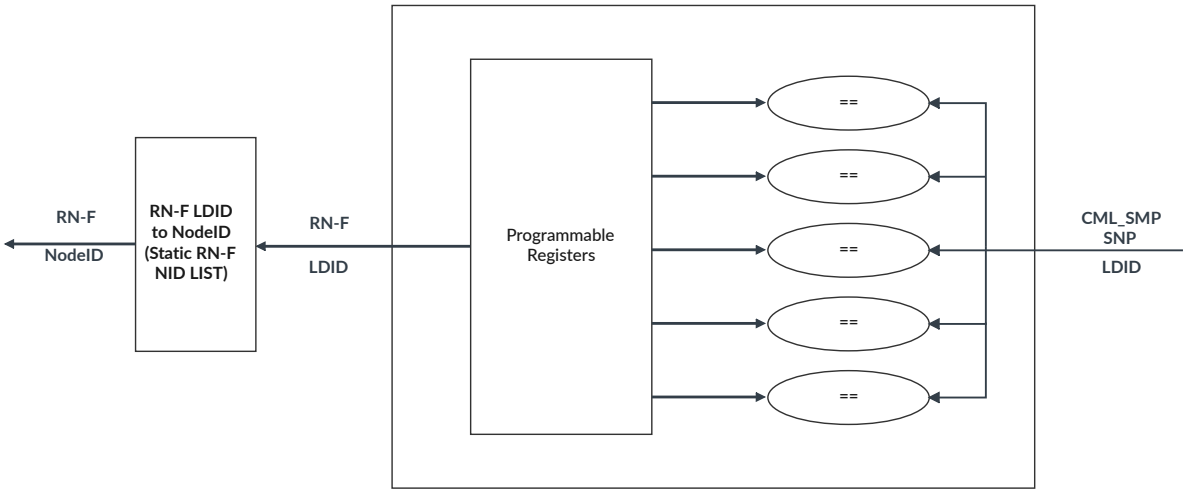
The CML-HA uses the LDID from incoming CHI snoop to perform a content match against the entries of programmable CML RAID to local LDID LUT. This content match results in the CML RAID sending a CML SMP snoop, as the following figure shows.

Figure 3-71: CHI SNP LDID to CML SNP RAID flow



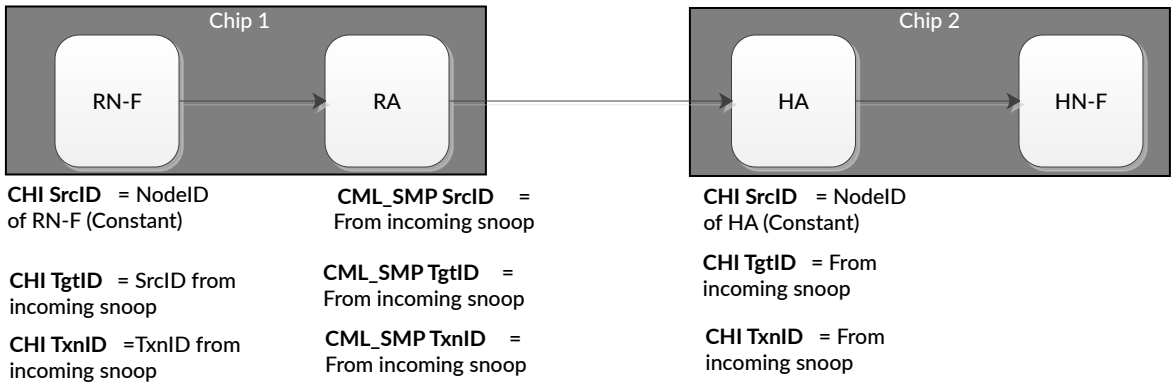
The following figure shows the detailed flow of a CHI SNP LDID to CML SNP RAID conversion.

Figure 3-72: CML SNP RAID to CHI RN-F LDID flow



The following figure shows all generated IDs used to route a snoop response from a remote RN-F on chip 1 to an HN-F on chip 2.

Figure 3-73: Remote RN-F to HN-F with all IDs generated



3.4.14 CMN-700 expanded RAID

In a CMN-700 based multi-chip system all the requesting agents are identified using a globally unique 10-bit *Requesting Agent ID* (RAID). This ID is made up of a 6-bit base ID and a 4-bit expanded ID.

The base ID component of the RAID must be unique across chips i.e., two agents on different chips cannot have the same base ID. For example, consider a configuration where one chip is assigned the base ID [5:0] = 0x4. In this case, no other chip in the system can have RAIDs with the base ID [5:0] = 0x4. This is important because some of the ID look up tables as well as non-protocol

messages use this 6-bit base ID for identification and routing. The 4-bit expansion ID enables CMN-700 to support up to 512 Requesting Agents (RAs) in a system.

Requesting agent on CMN-700 could be a caching agent like RN-F or an I/O coherent agent like RN-I and RN-D. Additionally, CMN-700 may have CXL Type1 devices attached to it and each of these Type1 devices will present one requesting agent to CMN-700. CXL Type1 requesting agent should be treated like a local caching agent in terms of RAID assignment, where it should be given a unique ID from the ID space of that given chip. On an SMP link, caching agents and I/O coherent agents can be assigned IDs from the same ID space. For e.g., an RN-I and RN-F can have the same ID. They must be unique within RN-Fs or RN-Is but not across. Further, PCIe RN-I and RN-Ds must get unique RAID within 7-bit RAID space as this 7-bit ID is used for “write gathering” at remote HN-P. This puts additional restrictions on the number of unique base IDs on a given SMP link.

Below are ID assignment rules that apply to CMN-700 systems:

- Identify all the CMN-700 caching agents that can communicate with the remote home. In a non-hierarchical caching system this would include all the local RN-Fs and CCG-HAs with CXL Type1 device attached.
- Identify all non-PCIe I/O coherent request agents (RN-Is and RN-Ds) that can communicate with the remote home.
- Identify all the PCIe RN-I and RN-D that can communicate with remote HN-P
- Take the greater of caching agents and non-PCIe I/O coherent agents and divide it by 16 to get the number of base IDs needed for non-PCIe agents.
- Take the number of PCIe RN-I and RN-Ds that can communicate with remote HN-P and divide it by 2 to get the number of base IDs needed for PCIe RN-I and RN-Ds
- Take the greater from step 4 and 5 to determine the number of base IDs needed per chip
- Assign a 10-bit unique ID to each of the caching agents using 6-bit base ID and 4-bit expanded ID.
- Assign a 10-bit unique ID to each of these non-PCIe I/O coherent requesting agents using 6-bit base ID and 4-bit expanded ID.
- Assign 7-bit unique ID to each of PCIe RN-I and RN-D using 6-bit base ID and 1-bit of expanded ID. If the number of PCIe RN-I and RN-Ds are greater than that can fit in 7-bit ID space then, for the remaining, assign 10-bit unique ID using 4-bit of expanded ID but the “write gathering” functionality must be disabled at these PCIe RN-I and RN-D. Write gathering at RN-I and RN-Ds is disabled by setting “dis_tnl_burst_early_dealloc_opt” bit in RN-I and RN-Ds Aux_ctl register por_<rni/rnd>_aux_ctl


Note

The above is applicable to PCIe RN-I and RN-Ds that can target remote HN-P. For PCIe RN-I and RN-Ds that do not target remote HN-P, 10-bit unique ID can be assigned using 6-bit base ID and 4-bit expanded ID

Example 3-7: RAID assignment in an example mixed CML system

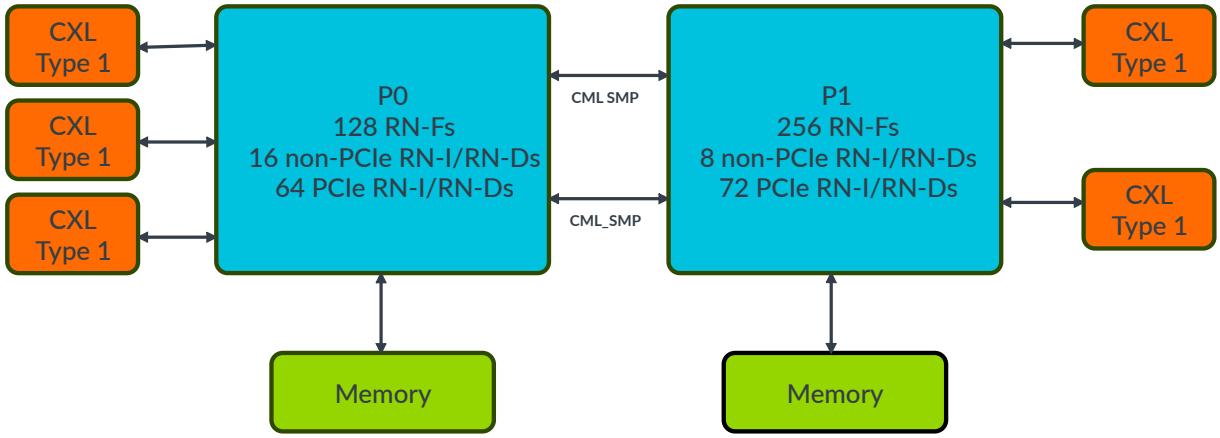
The following figure CXL system example, shows a system configuration containing two CMN-700 chips, P0 and P1.

The chips have the following characteristics:

P0 Chip is built using CMN-700, and contains 128 RN-Fs, 3 CXL Type1 devices attached, 16 non-PCIe RN-I/Ds and 64 PCIe RN-I/Ds

P1 Chip is built using CMN-700, and contains 256 RN-Fs, 2 CXL Type1 devices attached, 8 non-PCIe RN-I/Ds and 72 PCIe RN-I/Ds

Figure 3-74: CML system example



RAIDs across P0 and P1 are assigned according to the following scheme:

- P0 has 128 RN-Fs and 3 CXL Type1 attached devices. Which means that, in a non-hierarchical caching system, P0 has 131 caching agents. P0 has 16 non-PCIe RN-I and RN-Ds and 64 PCIe RN-I and RN-Ds. To get the number of unique base IDs required for non-PCIe agents, divide number of caching agents (greater of 131, 129, and 16) by 16, which gives 9. To get the number of unique base IDs required for PCIe RN-I and RN-Ds, divide number of PCIe RN-I and RN-Ds by 2, which gives 32. Since the greater of two is 32, P0 requires a minimum of 32 unique base ID values
- Assign base IDs 0 to 31 to P0. Toggle the expanded 4-bits to assign unique IDs to each of the caching agents. Similarly assign unique IDs to each of non-PCIe RN-I and RN-D using 0 to 31 base ID space and 4-bit of expanded ID. For PCIe RN-I and RN-Ds, assign unique IDs using 0 to 31 base IDs and bit [6] of expanded ID.
- P1 has 256 RN-Fs and 2 CXL Type1 attached devices. P1 has 8 non-PCIe RN-I and RN-Ds and 72 PCIe RN-I and RN-Ds. Similar to P0 calculation, P1 requires a minimum of 36 unique IDs from the base ID space. But since there are only 32 base IDs left, that's the max that can be assigned to P1
- Assign base IDs 32 to 63 to P1. Toggle the expanded 4-bits to assign unique IDs to each of the caching agents. Similarly assign unique IDs to each of non-PCIe RN-I and RN-D using 32 to 63 base ID space and 4-bit of expanded ID. For PCIe RN-I and RN-Ds, assign unique IDs using 32 to 63 base IDs and bit [6] of expanded ID. This would give unique IDs to 64 PCIe RN-I and RN-Ds. To support this, assign unique IDs to 63 PCIe RN-I and RN-Ds For the remaining 7 PCIe RN-I and RN-Ds, assign a unique ID using 32 to 63 base IDs and bit [6] of the expanded ID and disable "write gathering" functionality at these RN-I and RN-Ds

For the preceding example system,

LDIDs in the CMN-700 device P0 can be assigned in the following way:

- 128 RN-Fs on P0 are assigned LDIDs from 0-127. These default LDIDs are assigned when CMN-700 is built
- Three CXL Type1 devices are assigned LDIDs from 128 – 130. These must be programmed in CCG-HAs RAID to LDID LUT
- LDID assignment for remote RN-Fs
    1. In a non-hierarchical caching system, 128 RN-Fs from P0 are assigned LDIDs from 258-385. Three CXL Type1 devices are assigned LDIDs 386-388. These must be programmed in CCG-HAs RAID to LDID LUT
    2. In a hierarchical caching system, the entire chip (P0) represents as a caching agent and is assigned one LDID 258. 128 RN-Fs from P0 are assigned LDIDs from 259-386. These must be programmed in CCG-HAs RAID to LDID LUT

LDIDs in the CMN-700 device P1 are assigned in the following way:

- 256 RAs on P1 are assigned LDIDs from 0-255. These default LDIDs are assigned when CMN-700 is built
- Two CXL Type1 devices are assigned LDIDs from 256–257. These must be programmed in CCG-HAs RAID to LDID LUT
- LDID assignment for remote RN-Fs
    1. In a non-hierarchical caching system, 128 RN-Fs from P0 are assigned LDIDs from 258-385. Three CXL Type1 devices are assigned LDIDs 386-388. These must be programmed in CCG-HAs RAID to LDID LUT
    2. In a hierarchical caching system, the entire chip (P0) represents as a caching agent and is assigned one LDID 258. 128 RN-Fs from P0 are assigned LDIDs from 259-386. These must be programmed in CCG-HAs RAID to LDID LUT

3.5 Discovery

Discovery is a software algorithm that is used to discover the structure of the CMN-700 configuration, such as domains, components, and subfeatures while the system boots.

Software uses the discovery mechanism to identify the corresponding CHI node ID and logical ID for all of the following node types supported in CMN-700.

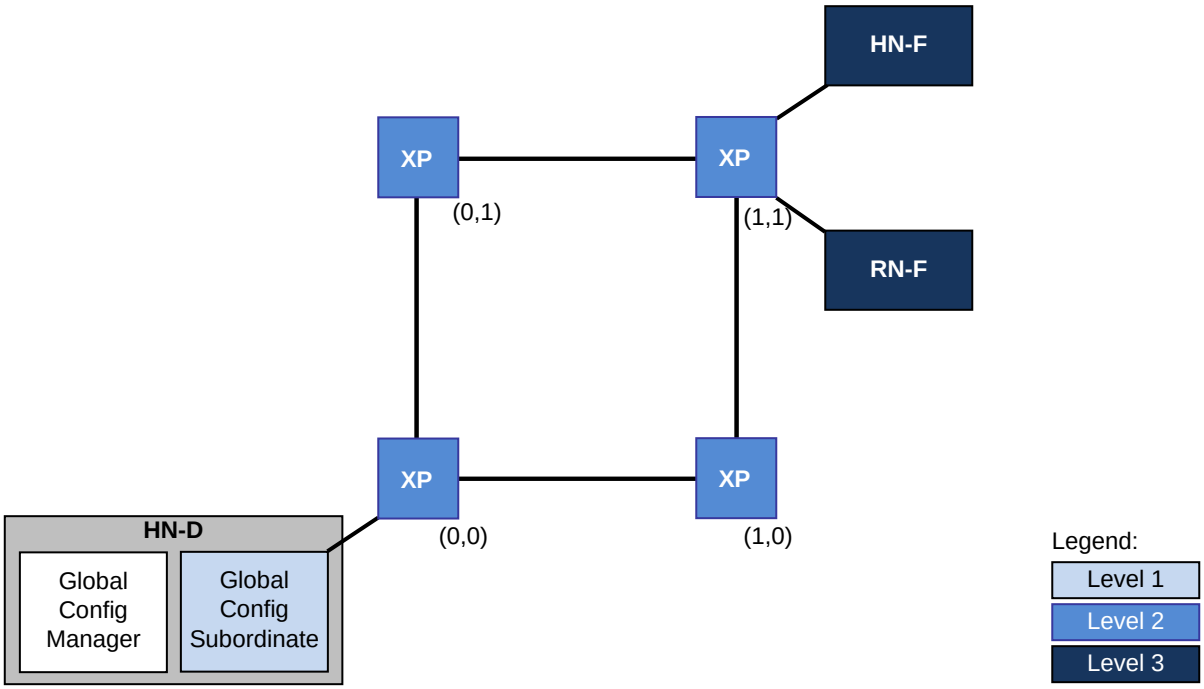
- DVM
- Global CFG
- DTC
- HN-F
- HN-F_MPAM_S
- HN-F_MPAM_NS

- HN-P
- HN-I
- MTU
- RN-D
- RN SAM
- RN-I
- SBSX
- XP
- CCG RA
- CCG HA
- CCLA

The following figure shows an example configuration. In the example, after discovery, software should have enough information to know the location of the following components:

- Global configuration registers
- Configuration registers for each XP
- Configuration registers for the HN-F
- Configuration registers for the RN SAM corresponding to the RN-F

Figure 3-75: 2 × 2 register tree example



3.5.1 Configuration address space organization

The configuration address space is organized depends on the system configuration. It is based on one system address, which is known as PERIPHBASE.

PERIPHBASE is the starting address of the range that all CMN-700 configuration registers are mapped to. For a CMN-700 system where both the X and Y dimensions are eight or less:

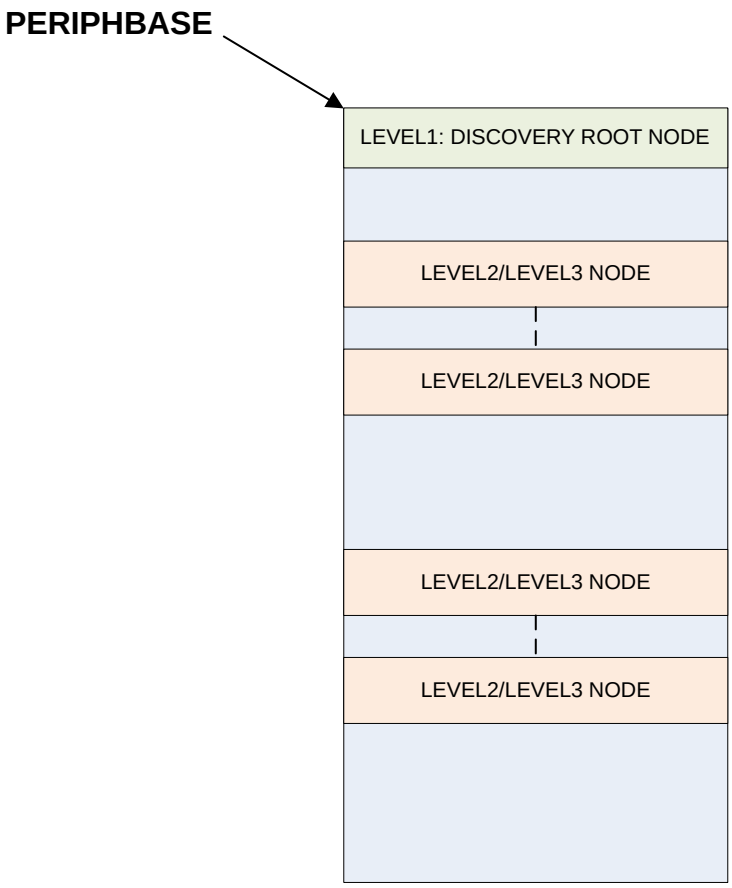
- This address must be aligned to 256MB
- The maximum size of configuration address space is 256MB

For a CMN-700 system where either the X or Y dimensions are nine or more:

- This address must be aligned to 1GB
- The maximum size of configuration address space is 1GB

Discovery determines the specific addresses for individual system blocks that have **IMPLEMENTATION DEFINED** register spaces, as shown in the following figure.

Figure 3-76: PERIPHBASE address map



CMN-700 supports 4B and 8B software-accessible registers. Register organization consists of software using 32-bit and 64-bit register reads.

All registers are organized into several register blocks and are referred to as nodes. A node:

- Is a register block with the size of 64KB
- Is associated with a logical block in the design
- Has information and configuration for that block that is specific to the implementation

The different types of nodes present in CMN-700 are:

General Contains device information and has children
Leaf Contains device information, such as configuration data, but has no children
Pure hierarchy Has children but contains no device information

If a node has more than one child, the node provides:

- The number of children

- A pointer to each child


Note

You can also find the address offsets for each node and configuration register in the IP-XACT file that Socrates™ generates for your custom mesh. Socrates™ stores the IP-XACT file with the rendered RTL in your Socrates™ workspace.

3.5.2 Configuration register node structure

Read-only registers that are organized into several register blocks are referred to as nodes.

Nodes are aligned on 8B boundaries (64KB aligned). The required registers are:

Node Information register

Identifies the product or node type, and the CHI node ID.

Child Information register

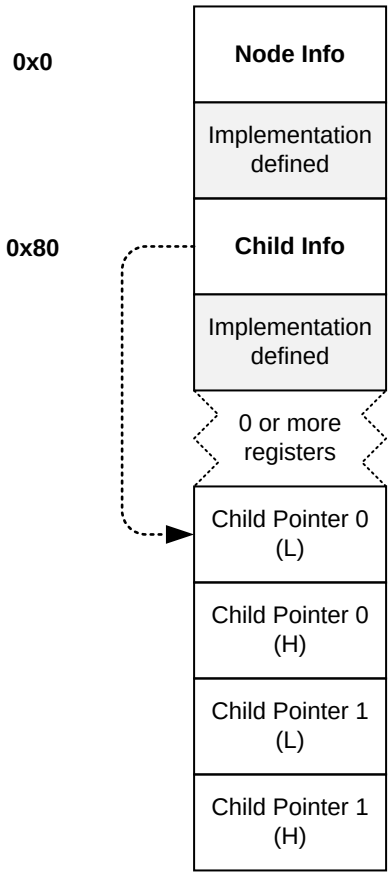
Indicates the child count and offset for the first register containing child node pointers. These optional Child Pointer registers each use 8B.


Note

The Node Information and Child Information registers are at fixed offsets for all nodes.

The following figure shows the basic node structure.

Figure 3-77: Basic node structure



The child_count field of the Child Information register indicates the number of children. This value also represents the number of functional units that are connected to the current unit on the next level of the discovery process.

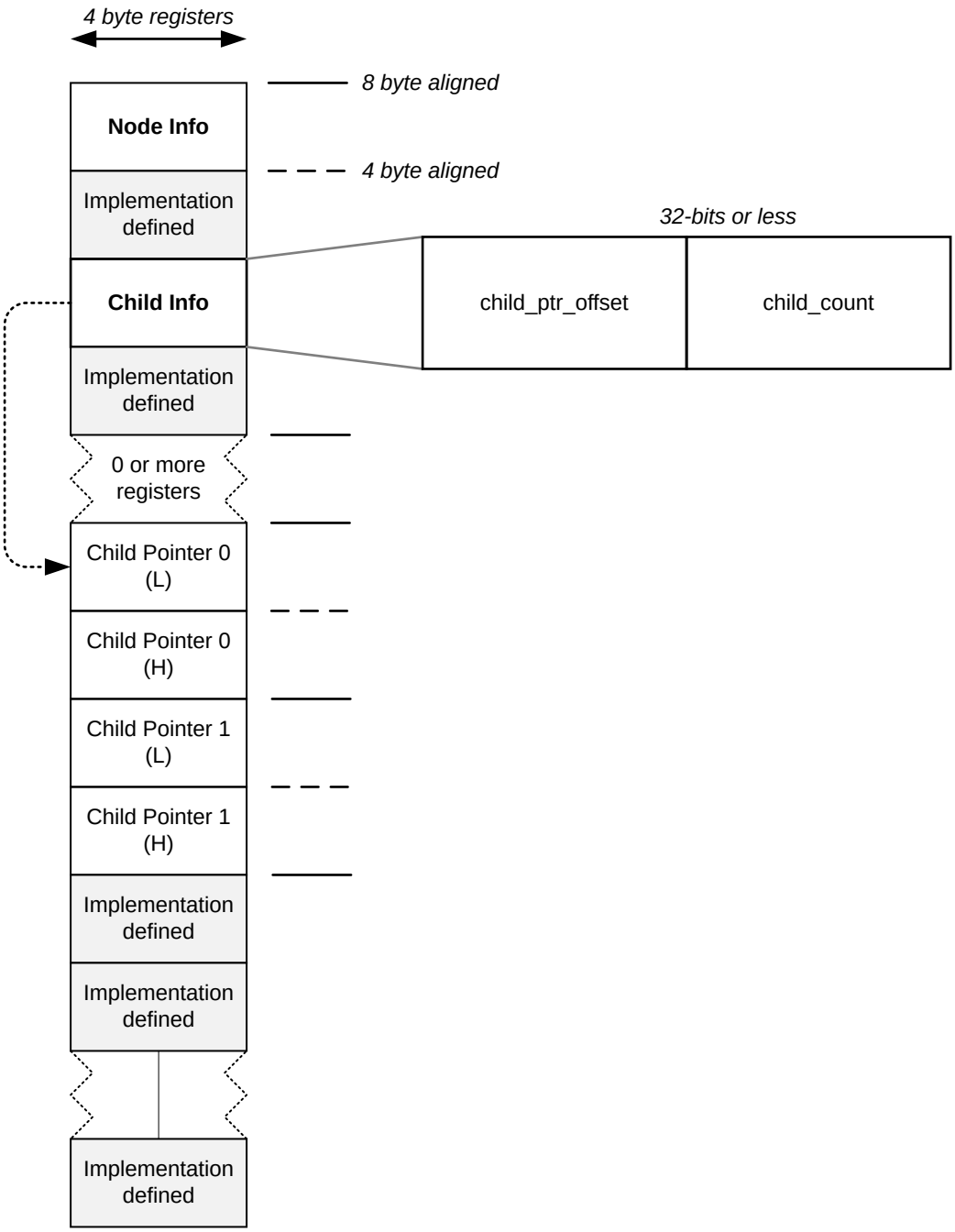
The child_ptr_offset field of the Child Information register indicates the Child Pointer 0 register offset, in bytes, from the Node Information register address.

Note

For a leaf node (node with no children), the child_count and child_ptr_offset fields are set to zero.

The following figure provides the node structure detail.

Figure 3-78: Node structure detail



The following table shows the supported node types and the corresponding node_type values in the Node Information register.

Table 3-58: node_type values

Node type	Value
Invalid	16'h0000

Node type	Value
DVM	16'h0001
CFG	16'h0002
DTC	16'h0003
HN-I	16'h0004
HN-F	16'h0005
XP	16'h0006
SBSX	16'h0007
HN-F_MPAM_S	16'h0008
HN-F_MPAM_NS	16'h0009
RN-I	16'h000A
RN-D	16'h000D
RN SAM	16'h000F
HN-P	16'h0011
CCG RA	16'h0103
CCG HA	16'h0104
CCLA	16'h0105
CCLA_RNI	16'h0106
APB	16'h1000

3.5.3 Child pointers

There is one child pointer register per child node.

The address of the register containing the first child pointer is computed as:

Base node address (of the current 64KB block) + the child_ptr_offset value (from the child_info register). Each subsequent child pointer register is 8 bytes higher. See Figure 3-78: Node structure detail on page 199.

For example:

- Base node address = 0x40000
- Child_ptr_offset in child info register = 0x100
- Address of first child pointer register (child pointer 0) = Base node address + child_ptr_offset = 0x40100
- Address to child pointer 1 = Address of child pointer 0 (0x40100) + 0x8 = 0x40108

Child pointers are 32 bits or less and are contained in the low register. The high register is zero. Child pointer contents include the following:

- The child node address offset from PERIPHBASE, bits 0-29, which is an unsigned integer (positive offset)

- One reserved bit, bit[30]
- An External Child Node indicator, bit[31]

For example, address to 64KB block of the child node = PERIPHBASE + child pointer register [29:0]. The child pointer register holds the child node address offset relative to PERIPHBASE.

The software performing the discovery can use two pieces of information:

1. The CHI node ID corresponding to the Config child node in question.
2. Information in the device port connection information register for the device port that the child node is connected to:
   a. por_mxp_device_port_connect_info_p0
   b. por_mxp_device_port_connect_info_p1
   c. por_mxp_device_port_connect_info_p2
   d. por_mxp_device_port_connect_info_p3


Note

By default, CMN-700 supports two device ports per MXP, P0 and P1. However, you can also extend the number of device ports. See 3.1.4.5 Support for extra device ports on MXPs on page 75.

The device type corresponding to that child node helps the discovery software determine if the child node is RN-F, RN SAM.

Depending on the size of the mesh (X and Y dimensions), CMN-700 supports three different widths for encoding the X and Y dimension. The number of bits that is required is selected based on the larger of the X and Y values.

Table 3-59: Mesh size and encoding bits

Mesh width in X dimension	Mesh width in Y dimension	Number of bits used to encode X, Y
X ≤ 4	Y ≤ 4	2 bits for X, 2 bits for Y
4 < X ≤ 8	Y ≤ 8	3 bits for X, 3 bits for Y
X ≤ 8	4 < Y ≤ 8	3 bits for X, 3 bits for Y
8 < X ≤ 10	8 < Y ≤ 10	4 bits for X, 4 bits for Y

3.5.4 Discovery tree structure

The one-time discovery process creates a lookup table that contains the addresses for all CMN-700 configured devices.

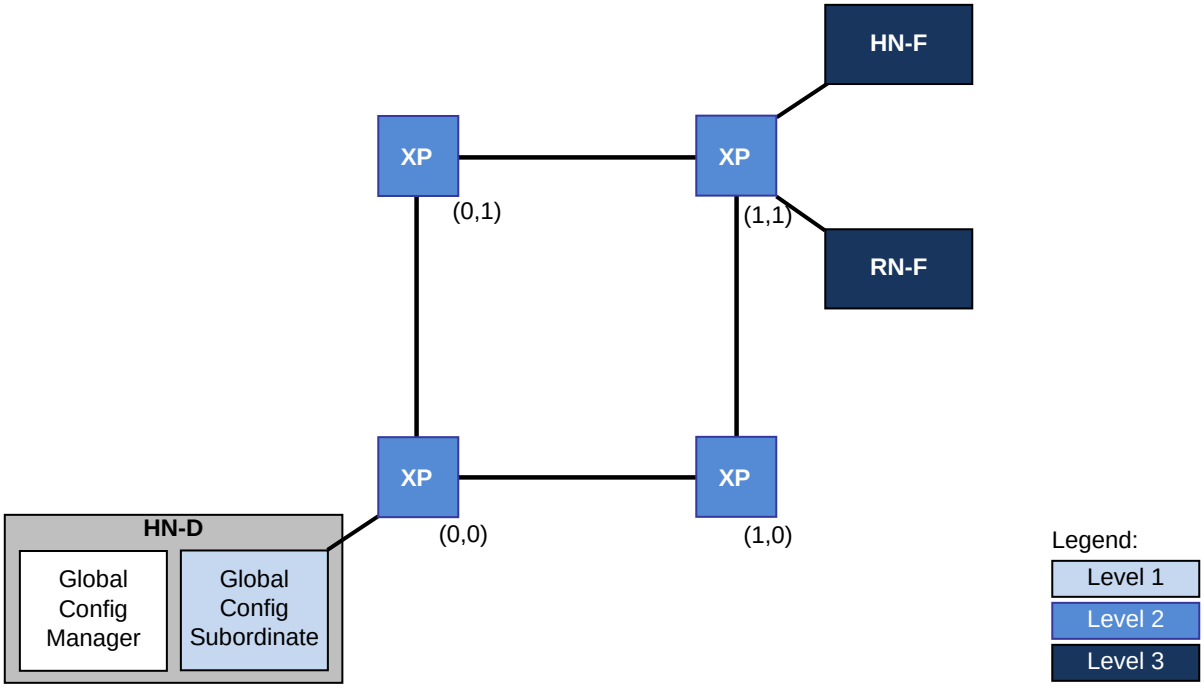
The discovery tree structure consists of three levels:

Level 1 Root Node, or the HN-D containing the Global Configuration Subordinate.
Level 2 XP layer.

Level 3 Leaf layer with one or two devices.

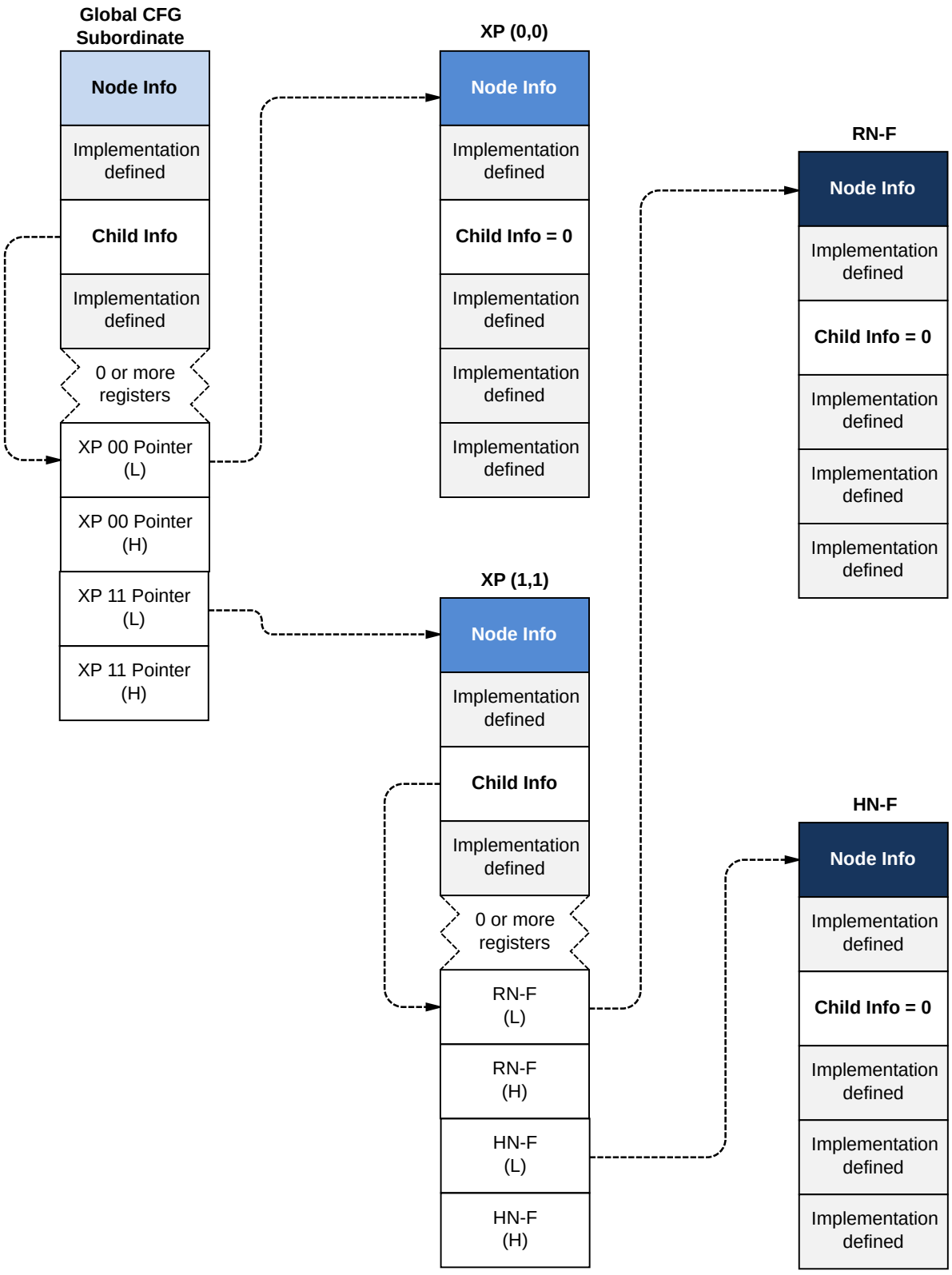
The following figure shows a 2 × 2 mesh configuration example with highlighted discovery tree levels.

Figure 3-79: 2 × 2 discovery tree example



The following figure shows the discovery tree structure for this 2 × 2 mesh configuration.

Figure 3-80: 2 × 2 discovery tree structure

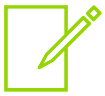


3.6 Link layer

CMN-700 provides link initialization, flow-control, and link deactivation functionality at the RN-F and SN-F device interfaces.

This functionality comprises the following mechanisms:

- A link initialization mechanism by which the receiving device communicates link layer credits, on each CHI channel that is present, to a transmitting device.
- A flow-control mechanism by which the transmitting device uses link layer credits to send CHI flits – one credit per flit. In turn, the receiving device sends these credits back to the transmitting device, one at a time, when it is done processing each flit to allow for subsequent flit transfers.


Note

The latency (in clock cycles) measured from the time a transmitting device uses a link layer credit to send a flit to the receiving device and the earliest time when it can receive that credit back from the receiving device and send a subsequent flit is called credit roundtrip latency.

- A link deactivation mechanism by which the transmitting device sends all unused link layer credits on each CHI channel back to the receiving device by sending corresponding link flits.

On flit upload channels, RN-F or SN-F is the transmitting device and CMN-700 is the receiving device. On flit download channels, CMN-700 is the transmitting device and RN-F or SN-F is the receiving device.

For a description of the functional requirements of the CHI link layer, see the *AMBA® 5 CHI Architecture Specification*.

3.6.1 Flit buffer sizing requirements

There are specific size requirements for the CMN-700 flit buffers.

Flit buffer sizing at a receiving device is based on the following two factors:

1. A transmitting device must be able to send flits continuously in a pipelined fashion without stalling because of insufficient link layer credits from the receiving device. This requirement ensures that the system can achieve the full link bandwidth. For a specific system, there is a minimum number of link layer credits that are required so that pipeline stalls can be prevented. You can use the credit roundtrip latency between the transmitting device and receiving device as a measure of the required number of link layer credits.
2. A receiving device must be able to accept and process as many flits as the number of link layer credits it has outstanding at the transmitting device. Therefore, the number of link layer credits that a receiving device sends must not exceed its flit buffering and processing capabilities.

Therefore, flit buffer sizing and corresponding link layer crediting must reflect the credit roundtrip latency. If this requirement is met, the system can achieve optimal flit transfer bandwidth between

transmitting and receiving devices. For more information about flit buffer sizing and link layer crediting for flit uploads and downloads at RN-F and SN-F interfaces, see the following sections:

- 3.6.2 Flit uploads from RN-F or SN-F on page 205
- 3.6.3 Flit downloads with RN-F or SN-F on page 205

3.6.2 Flit uploads from RN-F or SN-F

For flit uploads, the `RXBUF_NUM_ENTRIES` parameter specifies the number of flit buffers in CMN-700.

For more information about `RXBUF_NUM_ENTRIES`, see 2.5 Configurable options on page 22.

For optimal flit transfer bandwidth, this parameter must be set equal to the upload credit roundtrip latency (UpCrdLat<ch>) which is computed using the following equation.

UpCrdLat<ch> = UpCrdLatInt<ch> + UpCrdLatExt<ch>, where:

- <ch> is the CHI channel (REQ, RSP, SNP, or DAT).
- UpCrdLatInt<ch> is the upload credit latency inside CMN-700. This latency is measured, in clock cycles, from the time that the RN-F or SN-F asserts the RX<ch>FLITV input for a flit uploaded to CMN-700 to the earliest time when CMN-700 asserts the RX<ch>LCRDV output to the RN-F or SN-F after the flit is processed and the credit sent back. At the RN-F/SN-F interfaces, UpCrdLatInt<ch> = 1 on all CHI channels.
- UpCrdLatExt<ch> is the upload credit latency outside CMN-700. This latency is measured, in clock cycles, from the time that CMN-700 asserts the RX<ch>LCRDV output is when the credit is sent back to the RN-F or SN-F to the earliest time when the RN-F or SN-F asserts the RX<ch>FLITV input when the credit is used to send a subsequent flit.

3.6.3 Flit downloads with RN-F or SN-F

For optimal flit downloads, the RN-F or SN-F must size its input buffers to reflect the download credit roundtrip latency (DnCrdLat<ch>).

DnCrdLat<ch> is computed using the following equation.

DnCrdLat<ch> = DnCrdLatInt<ch> + DnCrdLatExt<ch>, where:

- <ch> is the CHI channel (REQ, RSP, SNP, or DAT).
- DnCrdLatInt<ch> is the download credit latency inside CMN-700. This latency is measured (in clock cycles) from the time RX<ch>LCRDV input is asserted by the RN-F or SN-F to CMN-700 to the earliest time when RX<ch>FLITV output is asserted by CMN-700 to the RN-F or SN-F for a flit using that credit. At the RN-F or SN-F interfaces, DnCrdLatInt<ch> = 2 on all CHI channels.
- DnCrdLatExt<ch> is the download credit latency outside CMN-700. This latency is measured (in clock cycles) from the time RX<ch>FLITV output is asserted by CMN-700 for a flit downloaded to the RN-F or SN-F to the earliest time when RX<ch>LCRDV input is asserted when the corresponding credit is returned by the RN-F or SN-F to CMN-700.

3.7 PCIe integration

CMN-700 supports integration of a PCIe *Root Complex* (RC) or *EndPoint* (EP).

3.7.1 PCIe topology requirements

There are specific topology rules that you must follow when integrating PCIe components with CMN-700.

The following PCIe topology requirements apply:

- PCIe subordinates must not be connected to HN-D.
- PCIe subordinates must not share HN-I with other non-PCIe subordinates.
- HN-P must only be used to connect to PCIe subordinates.

3.7.2 PCIe manager and subordinate restrictions and requirements

There are restrictions on both the types and flow of transactions between PCIe devices and CMN-700.


Note

In this section, PCIe HN-I refers to an HN-I or HN-P which has a PCIe subordinate that is connected to it. HN-I refers to all other HN-Is.

CMN-700 supports peer-to-peer PCIe traffic. For CMN-700, this is defined as a request from an RN-I or RN-D node with its pcie_mstr_present bit set targeting an HN-P or HN-I node, local or remote, with a PCIe device downstream. This function allows one PCIe endpoint to communicate with another PCIe endpoint through the interconnect.


Note

Peer-to-peer PCIe writes to a remote HN-P must use the tunneling flow when more than 1 CCG is present in CMN-700 to preserve write ordering. As such, *address based port aggregation* (CPAG) must be disabled for this traffic.

Transaction type restrictions

A PCIe manager must not send any *Non-Posted Configuration and I/O Writes* (NPR-Wr) targeting CMN-700.

Flow control requirements from CMN-700 to PCIe subordinate

The PCIe subordinate must be able to sink at least one NPR-Wr from CMN-700 sent on the PCIe HN-I AXI/ACE-Lite manager port. This requirement guarantees that the PCIe HN-I AW channel remains unblocked. Therefore, *Posted Writes* (P-Wrs) from PCIe manager targeting the downstream subordinate device can progress, as required by the PCIe ordering rules.

Flow control requirements from PCIe manager to CMN-700

Your configuration might have a *System Memory Management Unit* (SMMU) or GIC-ITS in the path between the PCIe manager interface and the RN-I subordinate interface. In this case, you must use one of the following mutually exclusive flow control options:

- Use a separate manager interface port in the SMMU and GIC-ITS for translation table walks (TCU in MMU-600 or GIC-600 and beyond). You can then connect this port to a different RN-I which does not send any requests to any PCIe HN-I. None of the managers that are connected to the RN-I can talk to any PCIe HN-I.
- When per port reservation is enabled use multiple AXI ports within the same RN-I or RN-D to connect SMMU or GIC-ITS for translation table walk. Per port reservation is described in the following section.

Enable per port reservation by clearing the dis_port_token of the por_{rni, rnd}_cfg_ctl register to 0. This programming enables reservation for all ports and also for read and write channels.

When per port reservation is enabled, each port has at least one reserved entry in both read and write transaction trackers. The per port reservation guarantees progress of requests through each port. You can increase the number of reserved entries according to your bandwidth requirements by programming the following register fields:

- s<X>_rd_token field of the por_{rni, rnd}_s<X>_port_control register
- s<X>_wr_token field of the por_{rni, rnd}_s<X>_port_control register

See 4.3.14.8 por_rnd_s0-2_port_control on page 977 and 4.3.15.8 por_rni_s0-2_port_control on page 1000.


Note

Program the field s{x}_tablewalk_mstr_present in config register por_{rni, rnd}_s{x}_port_control (x=0,1,2) for the corresponding port where the SMMU or GIC table walker is connected.

3.7.3 System requirements for PCIe devices

There are certain system-level requirements that you must meet when integrating PCIe devices with CMN-700. These requirements determine which CMN-700 devices can handle certain request types and how PCIe and non-PCIe transactions must be handled.


Note

In this section, PCIe HN-I refers to an HN-I or HN-P which has a PCIe subordinate connected to it. HN-I refers to all other HN-Is.

CMN-700 has the following system requirements for PCIe devices:

- All non-PCIe I/O subordinate devices must complete all writes without creating any dependency on a transaction in the PCIe subsystem.

- Your configuration might have an SMMU in the path between the PCIe manager interface and the RN-I subordinate interface. If using this kind of configuration, table-walk requests from the SMMU can only be sent to memory through the HN-F or non-PCIe HN-I.
- Interrupt translation table walk requests from GIC-ITS can only be sent to memory through the HN-F or non-PCIe HN-I.

There are certain programming requirements that your system must meet to ensure proper PCIe functionality. For more information, see 4.4.6 RN-I and HN-I PCIe programming sequence on page 1133.

3.8 Reliability, Availability, and Serviceability

The CMN-700 *Reliability, Availability, and Serviceability* (RAS) features are implemented as set of distributed logging and reporting registers and a central interrupt handling unit.

The distributed logging and reporting registers are associated with devices that can detect errors. These devices are XP, HN-I, HN-F, SBSX, and CCG

The central interrupt handling unit is located in the HN-D.

Each device that can detect errors logs the errors in local registers. The device sends error information to the central interrupt handling unit in the HN-D. The HN-D contains four sets of five error groups, which are based on the device type of the error source. The sets consist of a Secure and Non-secure group for errors, and a Secure and Non-secure group for fault-type errors. The groups are represented by *ERRor Group Status Registers* (ERRGSRs).

Each device type has up to 16 ERRGSRs, depending on how many devices of that type are present in the CMN-700 system. For example, the following table shows a possible configuration of the MXP ERRGSRs.

Table 3-60: Example MXP ERRGSR configuration

ERRGSR name	Register offset	Error group
por_cfgm_errgsr_mxp_0	16'h3000	MXP_<63:0> error status
por_cfgm_errgsr_mxp_1	16'h3008	MXP_<63:0> fault status
por_cfgm_errgsr_mxp_2	16'h3010	MXP_<127:64> error status
por_cfgm_errgsr_mxp_3	16'h3018	MXP_<127:64> fault status
por_cfgm_errgsr_mxp_0_NS	16'h3040	MXP_<63:0> error status NS
por_cfgm_errgsr_mxp_1_NS	16'h3048	MXP_<63:0> fault status NS
por_cfgm_errgsr_mxp_2_NS	16'h3050	MXP_<127:64> error status NS
por_cfgm_errgsr_mxp_3_NS	16'h3058	MXP_<127:64> fault status NS

If CMN-700 has ≤64 MXPs, only por_cfgm_errgsr_mxp_0, por_cfgm_errgsr_mxp_1, por_cfgm_errgsr_mxp_0_NS, and por_cfgm_errgsr_mxp_1_NS are present.

Each device that can detect errors has five Error Record registers that contain the error type, along with other information such as the address and opcode. Error types include *Corrected Error* (CE), *Deferred Error* (DE), and *Uncorrected Error* (UE).

For more information about error types, see 3.8.1 Error types on page 210.

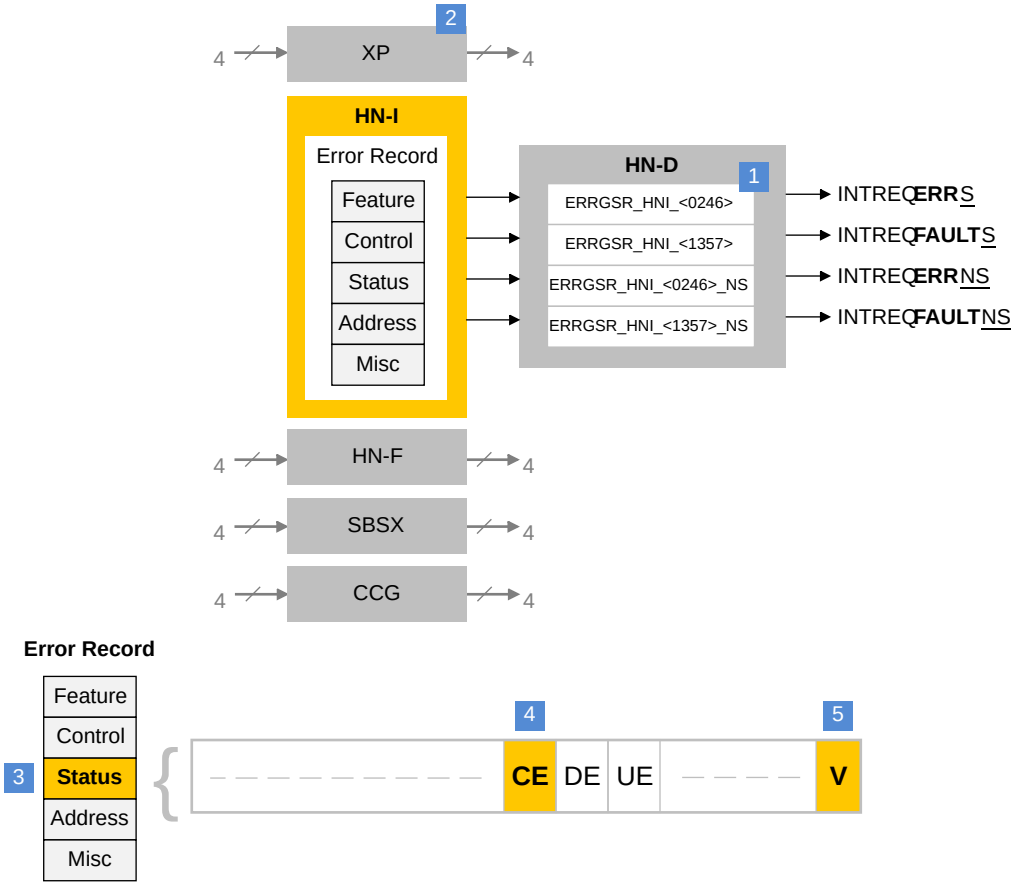
For register details, see 4.3 Register descriptions on page 284.

Error interrupt handler flow example

The following sequence of events and figure describe the process for determining the error source and type of an HN-I generating an interrupt request:

1. The HN-D generates an interrupt for one of the five error group types.
2. The error group indicates the error source device type, which can be:
   - XP
   - HN-I, which is used in this case
   - HN-F
   - SBSX
   - CCG
3. The bit location within the error group indicates the logical ID of that device type. In this case, it reveals an HN-I error, the HN-I Error Record Status block for this example.
4. The Status block of the Error Record for the specific XP, HN-I, HN-F, SBSX, or CCG indicates the type of error.
5. The Address and Misc blocks of the Error Record provide further details regarding error root cause, in this case a Corrected Error.
6. The Valid bit is also asserted.
7. To clear the asserted interrupt on the pin, the valid bit of the error status has to be cleared.

Figure 3-81: Error interrupt handler flow example



3.8.1 Error types

CMN-700 supports several error types.

The supported errors are:

- *Corrected Error* (CE)
- *Deferred Error* (DE)
- *Uncorrected Error* (UE)

Note

CEs, DEs, and UEs can occur simultaneously.

There might be cases when an error occurs and sets the status register, however, the interrupt reporting is not enabled. For other cases where the interrupt asserts, the interrupt request is

generated immediately when enabled. Otherwise, if interrupt reporting is disabled, any interrupt is cleared and the error remains logged with UE, DE, and CE.


Note

If both ERRCTLR.UI (uncorrected interrupt) and ERRCTLR.FI (fault interrupt) are set and a UE occurs, both fault and error interrupts are delivered from CMN-700.

Correctable Errors

Single-bit *Error Correcting Code* (ECC) errors can be corrected using ECC or other methods. The system handles these errors by completing the following steps:

1. Detects the error and increments the ERRMISC.CEC counter. Sets ERRSTATUS.AV and ERRSTATUS.MV. Logs the attributes in the ERRADDR and ERRMISC registers.
2. If CEC counter overflowed, the system updates ERRSTATUS.V and ERRSTATUS.CE, and sets ERRMISC.CECOF
3. Masks signaling of the error to the *RAS Control Block* (RCB) using ERRCTLR.CFI
4. If there are multiple CEC overflows, then the system sets ERRSTATUS.OF

Deferred Errors

These errors are UEs that have the following properties:

- Detected in one node of CMN-700, but the data is not used within the same node
- Poison bits are set for the data

The errors can be either fatal or non-fatal errors as described in this section. These errors are not correctable, but the system can operate for a time without being corrupted. These errors can be contained and the system might be able to recover using software means. They include:

- A data double bit ECC error in the SLC Data RAM
- Data check error detected in SLC

The system handles these errors as follows:

1. Logs the error information in the applicable ERRSTATUS, ERRADDR, and ERRMISC registers
2. Sets ERRSTATUS.V and ERRSTATUS.DE
3. Masks signaling of the error to the RAS control block using ERRCTLR.FI and ERRCTLR.UI
4. If there are multiple DEs, then the system sets ERRSTATUS.OF

Uncorrectable Fatal Errors

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

- A double-bit ECC error in SLC tag
- Flit parity error

- *Non-Data Error* (NDE) in a response packet

The system handles these errors as follows:

1. Logs the error information in the applicable ERRSTATUS, ERRADDR, and ERRMISC registers
2. Sets ERRSTATUS.V and ERRSTATUS.UE
3. Masks signaling of the error to the RAS control block using ERRCTLR.UI
4. If there are multiple UEs, then the system sets ERRSTATUS.OF

A component might not respond to further messages after an error is signaled. In this case, for the error handling routine to be successful, the component must still respond to configuration access requests from the configuration bus.


Note

When a subsequent error updates ERRSTATUS register, if the previous error has AV set, but the new error does not have address associated with it, the AV is not cleared.

CMN-700 follows the *Arm® Reliability, Availability, and Serviceability (RAS), for A-profile architecture* for mapping of the different error types to the interrupt. The following table summarizes the mapping of various error types.

Table 3-61: Mapping of error types

Interrupt type	Error type		
	Uncorrected error	Detected error	Corrected error
Fault Handling Interrupt	Yes, if ERRCTLR.FI==1	Yes, if ERRCTLR.FI==1	Yes, if ERRCTLR.CFI==1
Error Recovery Interrupt	Yes, if ERRCTLR.UI==1	No	No

3.8.2 Error Detection and Deferred Error values

For XP, the default values of *Error Detection* (ED) and DE depend on build time parameters.

Only the HN-F has CE counters that are implemented in the ERRMISC register. The default values of UI, FI, and CFI are `2'b10`, which enables control for the interrupt generation. The following table contains default values of ED and DE for XP.

Table 3-62: Default values of ED and DE for XP

POR_FLIT_PAR_EN	POR_DATACHECK_EN	MXP_DEV_DATACHECK_EN		ED	DE
		P0	P1		
0	0	0	0	`2'b00`	`2'b00`
0	1	0	0	`2'b01`	`2'b01`
		0	1	`2'b01`	`2'b01`
		1	0	`2'b01`	`2'b01`

POR_FLIT_PAR_EN	POR_DATACHECK_EN	MXP_DEV_DATACHECK_EN		ED	DE
		P0	P1		
		1	1	2'b00	2'b00
1	0	0	0	2'b01	2'b00
1	1	0	0	2'b01	2'b01
		0	1	2'b01	2'b01
		1	0	2'b01	2'b01
		1	1	2'b01	2'b00

For SBSX, if the AXDATAPOISON_EN parameter is not set, the default values of ED and DE are 2'b01. If the parameter is set, the default values are 2'b00.

For HN-I and HN-F, the default values of ED and DE are always 2'b01.

All fields are required, even though only HN-F has CE counters that are implemented in ERRMISC.

The default values of UI, FI, and CFI must be 2'b10, indicating that the interrupt generation is controllable.

3.8.3 Error detection, signaling, and reporting

Each CMN-700 component that is connected to a configuration bus can be included in the local error reporting mechanism.

The error handling protocol is as follows:

- Error overflow
- ERRSTATUS.OF value after errors
- ERRMISC fields and register bits

Error overflow

The overflow is set when different types of errors are detected. It is also asserted when multiple errors of equal priority are detected.

0b1 More than one error has been detected.
0b0 Only one error of the most significant type that ERRSTATUS.{UE, CE, DE} describes has been detected.

This bit is read/write-one-to-clear.


Note

ERRSTATUS.OF is only for the highest priority error. For example, if another DE follows the first DE, ERRSTATUS.OF is set. When the next UE happens, ERRSTATUS.OF is cleared. ERRSTATUS.OF is cleared because UE is the highest priority error in the system, and is the first occurrence of UE.

The following table shows the value of ERRSTATUS.OF after errors occur at t0, t1, and t2.

Table 3-63: ERRSTATUS.OF value after errors

Error at			Status of OF after error		
t0	t1	t2	t0	t1	t2
CE	CE	CE	0	1	1
CE	CE	DE	0	1	0
CE	CE	UE	0	1	0
CE	DE	CE	0	0	0
CE	DE	DE	0	0	1
CE	DE	UE	0	0	0
CE	UE	CE	0	0	0
CE	UE	DE	0	0	0
CE	UE	UE	0	0	1
DE	CE	CE	0	0	0
DE	CE	DE	0	0	1
DE	CE	UE	0	0	0
DE	DE	CE	0	1	1
DE	DE	DE	0	1	1
DE	DE	UE	0	1	0
DE	UE	CE	0	0	0
DE	UE	DE	0	0	0
DE	UE	UE	0	0	1
UE	CE	CE	0	0	0
UE	CE	DE	0	0	0
UE	CE	UE	0	0	1
UE	DE	CE	0	0	0
UE	DE	DE	0	0	0
UE	DE	UE	0	0	1
UE	UE	CE	0	1	1
UE	UE	DE	0	1	1
UE	UE	UE	0	1	1

ERRMISC fields

ERRMISC is the Secondary Error Syndrome Register. The fields of this register differ for ECC, parity, and other errors. The following table summarizes the valid fields for each unit.

Table 3-64: ERRMISC register bits

Bit	Component				
	XP	HN-I	HN-F	SBSX	CML HA
63	TLPMSG	-	CECOF	-	-
62	-		SETMATCH		
61			-		

Bit	Component				
	XP	HN-I	HN-F	SBSX	CML HA
60			ERRSET[12:0]		
59					
58	TargetID[10:0]				
57					
56		LPID[4:0]			ERRSET[7:0]
55					
54					
53					
52					
51		-			
50					
49		ORDER[1:0]			
48					
47	-	-	CEC[15:0]		-
46					
45					
44					
43					
42					
41					
40					
39					
38					
37					
36					
35					
34					
33					
32					
31			-		
30		SIZE[2:0]	MULTIWAYERR	SIZE[2:0]	
29			-		
28					
27		MEMATTR[3:0]	ERRWAY[7:0]	MEMATTR[3:0]	
26					
25					
24					
23		-		-	

<table>
<tr><th>Bit</th><th colspan="5">Component</th></tr>
<tr><th></th><th>XP</th><th>HN-I</th><th>HN-F</th><th>SBSX</th><th>CML HA</th></tr>
<tr><td>22</td><td rowspan="6">OPCODE[6:0]</td><td rowspan="6">OPCODE[6:0]</td><td rowspan="3"></td><td rowspan="4"></td><td rowspan="8"></td></tr>
<tr><td>21</td></tr>
<tr><td>20</td></tr>
<tr><td>19</td><td rowspan="3">-</td></tr>
<tr><td>18</td></tr>
<tr><td>17</td><td rowspan="2">OPTYPE[1:0]</td></tr>
<tr><td>16</td><td>OPTYPE</td></tr>
<tr><td>15</td><td rowspan="10">SRCID[10:0]</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>14</td><td rowspan="11">SRCID[10:0]</td><td rowspan="11">SRCID[10:0]</td><td rowspan="11">SRCID[10:0]</td></tr>
<tr><td>13</td><td rowspan="11">SRCRAID[9:0]</td></tr>
<tr><td>12</td></tr>
<tr><td>11</td></tr>
<tr><td>10</td></tr>
<tr><td>9</td></tr>
<tr><td>8</td></tr>
<tr><td>7</td></tr>
<tr><td>6</td></tr>
<tr><td>5</td></tr>
<tr><td>4</td><td rowspan="4">ERRSRC[4:0]</td></tr>
<tr><td>3</td><td rowspan="4">ERRSRC[3:0]</td><td rowspan="4">ERRSRC[3:0]</td><td rowspan="4">-</td><td>-</td></tr>
<tr><td>2</td><td rowspan="3">ERRSRC[2:0]</td></tr>
<tr><td>1</td></tr>
<tr><td>0</td></tr>
</table>

3.8.4 Error reporting rules

CMN-700 uses specific error reporting rules, concerning which errors must be reported and propagated.

The rules regarding error reporting in CMN-700 are:

- Any error originating in CMN-700 is reported
- Any error originating outside CMN-700 but corrupting CMN-700 is reported
- The HN-I can report an error in a response packet from outside CMN-700 if it does not propagate the response any further
- All non-posted write errors are propagated where possible

3.8.5 HN-F error handling

Errors are reported at the HN-F for various reasons.

Request errors at HN-F

The HN-F detects:

- ECC errors in SF Tag, SLC Tag, and Data RAMs.
- Data check and poison errors on DAT flits.
- *Non-Data Errors* (NDEs) on responses.
- Memory Address Decode Errors

ECC errors in SF Tag, SLC Tag, and Data RAMs

HN-F detects single-bit and double-bit ECC errors in the SF Tag, SLC Tag, and Data RAMs. It can correct single-bit ECC errors. Such errors are logged and reported as CEs.

The source of the double-bit ECC errors determines how they are handled.

SLC Data RAM

- Logged and reported as DEs.
- Propagated to the data consumer in the form of data poison.

SF Tag RAM

- Logged and reported as DEs.
- Not propagated to the requestor.
- The SF Tag RAM in the HN-F is disabled following the first occurrence of the double-bit ECC error.

SLC Tag RAM

- Fatal error.
- Logged and reported as UEs.
- Propagated to the requestor as NDEs in the responses.

Data check and poison errors on DAT flits

If data is allocated by the HN-F, the HN-F detects data check errors and poison error on the data flits. In such cases, HN-F logs and reports the data check error as a DE. If HN-F allocated the data in SLC Data RAM, it converts the data check error into data poison for all subsequent requests to this cache line.

If cmn_hns_aux_ctl.hnf_poison_intr_en == 1, then the poison errors originating from HN-F are logged and reported as UEs.

NDEs on responses

HN-F can receive NDEs from other data and response sources such as RN-F, RN-I, and SN-F. If the cache line was allocated in SLC Data RAM, it is logged and reported as a UE. If the cache line is not allocated in HN-F SLC, it propagates the errors to the requestor as an NDE. If SN-F is not CHI-E interface and can send mixed RESPERR in data responses. RN-F and RN-I needs to handle the error properly.

Memory Address Decode Error Handling

HN-F can receive requests and generate SLC and SF victims that HNSAM detects as an invalid memory region. One example is to support CXL hot-plugging memory. Upon detecting any memory address decode errors, HN-F completes transactions internally without sending any SN requests, and logs decode errors as UE/DE based on configuration register.

HN-F Configuration Registers:

- If poison_on_mem_addr_dec_err_en is set, HN-F returns all 0s in data with DAT.poison bits set for a read request that has Memory Address Decode Error. Otherwise, HN-F returns all 1s in data with all DAT.poison bits cleared
- If ue_on_mem_addr_dec_err is set, HN-F logs Memory Address Decode Error as UE. Otherwise, HN-F logs it as DE

3.8.6 HN-I error handling

Errors are reported at the HN-I for various reasons.

3.8.6.1 Request errors at HN-I

The HN-I detects errors on receiving various request types and sends an NDE response to the requesting RN.

The HN-I logs request information in the error logging registers, por_hni_erraddr(_NS) and por_hni_errmisc(_NS). They are marked as DEs in the error status register, por_hni_errstatus(_NS). The HN-I detects errors on receiving the following request types:

- Coherent Read.
- CleanUnique/MakeUnique.
- Coherent/CopyBack Write.
- Atomic.
- Illegal Configuration Read/Write (HN-D only).
- Unsupported Exclusive access (HN-P only).

The reqerr_cohreq_en configuration bit in the por_hni_cfg_ctl register enables or disables the sending of NDE responses and logging of error information for following request types. By default, this bit is enabled. It can only be programmed during boot time to any one of the following:

- Coherent Read.
- CleanUnique/MakeUnique.
- Coherent/CopyBack Write.

The HN-P has a configuration bit, the disable_hnp_excl_err bit, in the por_hni_cfg_ctl register. This bit disables the sending of NDE responses and logging of error information for unsupported Exclusive accesses. Exclusive WriteNoSnp and Exclusive ReadNoSnp requests are unsupported in HN-P as HN-P is not a PoS device. Disabling this error results in an Exclusive Pass response to these requests.

The following table lists all the requests that an HN-I detects as errors and the support of reqerr_cohreq_en.

Table 3-65: HN-I request errors and support for configuration bit

Request type	Reqerr_cohreq_en controls sending of NDE and log error
Coherent Read	Yes
CleanUnique/MakeUnique	Yes
Coherent/CopyBack Write	Yes
Atomics	No
Illegal Configuration Read/Write	No

- Coherent reads are downgraded to ReadNoSnp and sent downstream (AXI/ACE-Lite subordinate).
- Coherent/Copyback writes are downgraded to WriteNoSnp and sent downstream (AXI/ACE-Lite subordinate).
- Illegal Configuration Read is sent as ReadNoSnp to downstream (AXI/ACE-Lite subordinate).
- CleanUnique, MakeUnique, Atomics, and Illegal Configuration Writes are handled within HN-I.
- StashOnceShared, StashOnceUnique, and PrefetchTgt are completed within HN-I without any errors.

3.8.6.2 Data Errors at HN-I

The HN-I only detects errors on write data if it does not detect an error on that request.

The following provides an overview of AXI and ACE-Lite write requests and configuration write requests, with no request error:

- For AXI and ACE-Lite write requests with no request error, when they receive Poison error on data, the HN-I detects the error. If downstream does not support poison, the HN-I logs the request information in por_hni_erraddr(_NS) and por_hni_errmisc(_NS). The write requests are marked as UEs in the error status register, por_hni_errstatus(_NS).
- For configuration write requests with no request error, on receiving write data with Partial ByteEnable error, Data check error, or Poison, HN-I detects and sends an NDE response to the requesting RN. It logs the SrcID and TxnID of the request and drops the write. They are marked as DEs in the error status register, por_hni_errstatus(_NS).


Note

StashOnceShared, StashOnceUnique, and PrefetchTgt are completed within HN-I without any errors.

3.8.6.3 Response Errors at HN-I

The HN-I only detects errors on write response if it does not detect an error on that request.

To summarize:

- For AXI/ACE-Lite write requests with early completions from HN-I and no request error, on receiving *Subordinate Error* (SLVERR), or *Decode Error* (DECERR) on downstream write response (BRESP), HN-I detects the error. It logs request information in por_hni_erraddr(_NS) and por_hni_errmisc(_NS). They are marked as UEs in the error status register, por_hni_errstatus(_NS).
- For AXI/ACE-Lite write requests with downstream completions and no request error, SLVERR, or DECERR on downstream write response (BRESP) are passed on to the requesting RN as CHI DEs or NDEs.
- For AXI/ACE-Lite read requests, SLVERR and Poison (if supported by downstream) are both converted to Poison within the CMN-700 system, independent of error on request. DECERRs on downstream read responses are passed on to the requesting RN.

3.8.6.4 HN-I summary on sending NDE and DE

The HN-I sends NDE scenarios for certain situations.

The HN-I sends NDE in the following cases:

- Request Error.
  - Coherent Read, if reqerr_cohreq_en is set to 1
  - CleanUnique/MakeUnique, if reqerr_cohreq_en is set to 1
  - Coherent/CopyBack Write, if reqerr_cohreq_en is set to 1
  - Atomic
  - Illegal Configuration Read/Write, HN-D only
  - Unsupported Exclusive access, HN-P only


Note

For the legal format of configuration read/write request, see 4.1.5 Requirements of configuration register reads and writes on page 255.

- Write Data Error for Configuration Write request, HN-D only:
  - Partial ByteEnable Error
  - Data Check Error
  - Poison
- AXI/ACE-Lite Response Error:
  - DECERR on *downstream write response* (BRESP) for writes with downstream completions
  - DECERR on *downstream read response* (RRESP)

The HN-I sends DE in the following cases:

- AXI/ACE-Lite Response Error:
  - SLVERR on BRESP for writes with downstream completions

3.8.6.5 HN-I summary on logging errors

The HN-I logs an error as deferred or uncorrected in certain conditions.

Deferred errors

The HN-I logs an error as deferred in the following cases:

- Request error.
  - Coherent Read, if reqerr_cohreq_en is set to 1
  - CleanUnique/MakeUnique, if reqerr_cohreq_en is set to 1
  - Coherent/CopyBack Write, if reqerr_cohreq_en is set to 1
  - Atomic
  - Illegal Configuration Read/Write
  - Unsupported Exclusive access, HN-P only. This error type is disabled if disable_hnp_excl_err is set to 1.


Note

For the legal format of a Configuration Read/Write request, see 4.1.5 Requirements of configuration register reads and writes on page 255.

- Write Data Error for Configuration Write request:
  - Partial ByteEnable Error
  - Data Check Error
  - Poison Error

Uncorrected errors

The HN-I logs an error as uncorrected in the following cases:

- Write Data Error for AXI/ACE-Lite write requests:
  - Poison Error on data if downstream does not support poison
- AXI/ACE-Lite Write Response Error:
  - SLVERR or DECERR on BRESP for writes that were sent early completions

3.8.6.6 CML configuration with HN-I

In CML configuration, HN-I must be configured to report NDE response on coherent requests.

This requirement is met by setting por_hni_cfg_ctl.reqerr_cohreq_en.

3.8.7 SBSX error handling

The following describes how errors are handled at the SBSX.

If the AXI memory controller downstream of SBSX does not support POISON (indicated by por_sbsx_unit_info.axdata_poison_en = 0), and if CHI Write Data has Poison set, then SBSX detects and logs this error.

SBSX does not have opcode-based Request/Response Error class as does HN-I.

The following table shows the SBSX summary on sending an NDE/DE.

Table 3-66: SBSX summary on sending NDE/DE

Case number	Source of error	SBSX error response
1	Decode Error on RDATA from AXI side	NDE on COMP_DATA on CHIE side
2	Subordinate Error on RDATA from AXI side	Poison on COMP_DATA on CHIE side
3	Decode Error on BRESP from AXI side	NDE on COMP for CMOs or Writes with EWA=0
4	Subordinate Error on BRESP from AXI side	DE on COMP for CMOs or Writes with EWA=0

3.8.8 RN-I error handling

RN-I does not report any errors. When a parity error is detected in the *Read Data Buffer* (RDB) RAMs, RN-I or RN-D propagates the error on AXI R channel as RPOISON or RRESP.

When a parity error is detected in the *Write Data Buffer* (WDB) RAMs, RN-I, or RN-D propagates the error on CHI TXDAT POISON field.

3.8.9 XP error handling

Errors are reported at the XP for various reasons.

The following errors are detected in the XP:

- Flit parity error.
- Data check error (DAT channel only).

Flit parity error

Flit parity is generated on a flit upload from a device port to a mesh port, for both internal and external devices. Flit parity check is done on a flit download from a mesh port to a device port.
Flit parity is not generated or checked when a flit is bypassed or looped back across the device ports on the same XP.

Data check error

Data check is enabled in the XP using the `DATACHECK_EN` parameter.
Data check (Data Byte Parity) bits are generated corresponding to each byte of data on a DAT flit upload from a device port when the corresponding device does not support Datacheck (For RN-F nodes, this is indicated by `DEV_DATACHECK_EN` = 0).

Data check is accomplished on a flit download to a device which does not support Data check.

Data check bits are generated and checked when a DAT flit is bypassed or looped back across the device ports on the same XP when the corresponding devices involved do not support Data check.


Note

When the global `DATACHECK_EN` parameter is TRUE, the Check_Type property for all ACE-Lite interfaces becomes `Odd_Parity_Byte_Data`. This means, for example, HN-I and SBSX nodes will not check data byte parity for outbound write data, and will simply propagate the DAT.DataCheck value it received from CHI on the downstream `WDATACHK` signals.

Error reporting and logging

Flit parity and Data check errors are reported to the RCB. The following table contains flit fields that are logged in the XP configuration register.

Table 3-67: XP configuration register flit fields

Error source	Errstatus					Errmisc			
-	DE	CE	MV	UE	V	ERRSRC	SRCID	OPCODE	TargetID
Data Parity P0 REQ channel	1	0	1	0	1	`5'b00000`	v	v	v

Error source	Errstatus					Errmisc			
Data Parity P1 REQ channel	1	0	1	0	1	5'b00001	v	v	v
Data Parity P2 REQ channel	1	0	1	0	1	5'b00010	v	v	v
Data Parity P3 REQ channel	1	0	1	0	1	5'b00011	v	v	v
Data Parity P0 RSP channel	1	0	1	0	1	5'b01000	v	v	v
Data Parity P1 RSP channel	1	0	1	0	1	5'b01001	v	v	v
Data Parity P2 RSP channel	1	0	1	0	1	5'b01010	v	v	v
Data Parity P3 RSP channel	1	0	1	0	1	5'b01011	v	v	v
Data Parity P0 SNP channel	1	0	1	0	1	5'b10000	v	v	0
Data Parity P1 SNP channel	1	0	1	0	1	5'b10001	v	v	0
Data Parity P2 SNP channel	1	0	1	0	1	5'b10010	v	v	0
Data Parity P3 SNP channel	1	0	1	0	1	5'b10011	v	v	0
Data Parity P0 DAT channel	1	0	1	0	1	5'b11000	v	v	v
Data Parity P1 DAT channel	1	0	1	0	1	5'b11001	v	v	v
Data Parity P2 DAT channel	1	0	1	0	1	5'b11010	v	v	v
Data Parity P3 DAT channel	1	0	1	0	1	5'b11011	v	v	v
FLIT Parity P0 REQ channel	0	0	1	1	1	5'b00000	v	v	v

Error source	Errstatus					Errmisc			
FLIT Parity P1 REQ channel	0	0	1	1	1	5'b00001	v	v	v
FLIT Parity P2 REQ channel	0	0	1	1	1	5'b00010	v	v	v
FLIT Parity P3 REQ channel	0	0	1	1	1	5'b00011	v	v	v
FLIT Parity P0 RSP channel	0	0	1	1	1	5'b01000	v	v	v
FLIT Parity P1 RSP channel	0	0	1	1	1	5'b01001	v	v	v
FLIT Parity P2 RSP channel	0	0	1	1	1	5'b01010	v	v	v
FLIT Parity P3 RSP channel	0	0	1	1	1	5'b01011	v	v	v
FLIT Parity P0 SNP channel	0	0	1	1	1	5'b10000	v	v	0
FLIT Parity P1 SNP channel	0	0	1	1	1	5'b10001	v	v	0
FLIT Parity P2 SNP channel	0	0	1	1	1	5'b10010	v	v	0
FLIT Parity P3 SNP channel	0	0	1	1	1	5'b10011	v	v	0
FLIT Parity P0 DAT channel	0	0	1	1	1	5'b11000	v	v	v
FLIT Parity P1 DAT channel	0	0	1	1	1	5'b11001	v	v	v
FLIT Parity P2 DAT channel	0	0	1	1	1	5'b11010	v	v	v
FLIT Parity P3 DAT channel	0	0	1	1	1	5'b11011	v	v	v

If the device supports Poison (indicated by DEV_POISON_EN = 1), the Datacheck error is factored in the POISON field of the DAT flit. Else, it is factored in as DataError in the RESPERR field.

3.8.10 CCG error handling

CCG reports errors for various reasons which includes

- Errors detected in CCG RAM structures
- On incoming CXL viral, if configured to report through CMN-700's error reporting mechanism.
- On incoming CXL poison

Contents of the RAM structures have byte parity protection and on any parity error, that is detected when the contents are read, are reported as either *Uncorrectable Error* (UE) or *Deferred Error* (DE).

The following fields are stored in RAMs:

- Data
- *Byte Enables* (BEs)
- Poison
- Metadata, 24-bits

DEs

- Parity errors on Data, Poison, and Metadata are reported and logged as DEs. If an error is detected on the Data field, then the poison bit for the corresponding 64-bit data chunk is set. If an error is detected on the Poison or Metadata fields, then all the poison bits are set.
- In CXL mode, any inbound poison notification on .cache or .mem data is reported as DE. This information can be used to determine the entry point of poisoned data in CMN mesh.

Table 3-68: Priority ranking for DE errors at CCG

DE type	Priority
CCLA Data RAM	1 (highest)
Passive Buffer Data RAM	2
Read Data Buffer bank0	3
Read Data Buffer bank1	4
Write Data Buffer bank0	5
Write Data Buffer bank1	6
CXL.mem	7
CXL.cache Poison	8

UEs

- A parity error on BEs is reported and logged as UE. In this case, all the BEs are driven to 0s to make sure that data is not written to the next level.
- In CXL mode, any inbound CXL viral notification can be reported as a UE. By default, this reporting is disabled but can be enabled by clearing the disable_viral_err_report bit in the CCLA AUX_CTL register (por_ccla_aux_ctl).

Table 3-69: Priority ranking for UE errors at CCG

UE type	Priority
CXL Viral	1 (highest)
Passive Buffer Data RAM	2
Write Data Buffer bank0	3
Write Data Buffer bank1	4

3.9 Transaction handling

The handling of certain CHI transaction types and fields differs according to the CMN-700 device type.

Some devices fully support certain transaction types or fields, whereas others do not do any processing of those transactions. Furthermore, some transaction types are unsupported, such as barriers.

3.9.1 Atomics

CMN-700 supports atomic accesses to both cacheable and non-cacheable memory locations.

3.9.1.1 Atomic requests in HN-F

The HN-F completes all CHI atomic requests that it receives, both for Cacheable and Non-cacheable transactions.

For Cacheable transactions, the HN-F completes any appropriate coherent actions and, if necessary, obtains the targeted cache line from memory. The HN-F then completes the required atomic operation and issues the appropriate response, with or without data.

For Non-cacheable transactions, the HN-F does not send an atomic request to the SN. As the final PoS/PoC for all memory traffic, the HN-F is able to issue a read to the SN, atomically update the copy of the data in the HN-F, and then write back the result to the SN. This approach means that the SN never receives CHI atomic requests, as the HN-F completely handles the requests.

3.9.1.2 Atomic requests in SN

The SN node (CHI memory controller or SBSX bridge) does not process atomic requests.

3.9.1.3 Atomic requests in HN-I

The HN-I does not complete atomic transactions.

On receiving an atomic request, the HN-I generates an appropriate error response to the originating manager.

3.9.1.4 Atomic requests in RN-I and RN-D

RN-I and RN-D support atomic transactions in CMN-700. These nodes can receive atomics from ACE5-Lite and AXI5 managers, and translate them on CHI before sending them to HN-S, HN-I, or CML nodes.

Atomics and write transactions share a write tracker for processing in RN-I and RN-D. There is a separate *Read Data Buffer* (RDB) for atomic responses. Atomic requests issue from RN-I or RN-D once both write tracker entry and Atomic RDB have been allocated. The Atomic RDB deallocates once the return read data is issued on the AXI R channel. The write tracker entry will then deallocate once both Bresp and Rresp for the atomic request have been sent. The `NUM_ATOMIC_BUF` parameter determines the depth of the Atomic RDB, and should be set to larger values for better atomic request performance, if needed.


Note

For atomic transactions arriving at RN-I and RN-D from ACE5-Lite and AXI5 managers, all write strobes within AWSIZE must be set. RN-I and RN-D do not allow sparse write strobes for atomic transactions.

3.9.2 Exclusive accesses

CMN-700 supports exclusive accesses to both Shareable and Non-shareable locations.

See the *AMBA® 5 CHI Architecture Specification*.

3.9.2.1 Exclusive accesses in HN-F

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

RNs generate ReadNoSnp and WriteNoSnp Exclusives for memory locations that are marked Non-cacheable or Device. ReadPreferUnique, MakeReadUnique, ReadShared, ReadClean, ReadNotSharedDirty, and CleanUnique exclusives are used for shareable and coherent memory locations.

Each HN-F in CMN-700 can support tracking of up to 512 logical processors for non-Cacheable exclusive operations:

- In configurations with up to 64 RN-Fs, HN-F supports 64 exclusive monitors.
- In configurations above 144 RN-Fs, the total number of exclusive monitors is equivalent to the total number of local and remote RN-Fs, RN-Is, and RN-Ds, up to a maximum of 512 exclusive monitors.

For non-Cacheable exclusive, the system programmer must ensure that there are no more logical processors capable of concurrently sending exclusive operations than the number of exclusive monitors.

Cacheable exclusive is implemented based on Snoop Filter and not constrained by the number of exclusive monitors.

3.9.2.2 Exclusive accesses in HN-I

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

Each HN-I partition includes exclusive monitors, as defined in the *AMBA® 5 CHI Architecture Specification*, for tracking of these transaction types. The number of monitors is specific to the configuration and is determined by the number of RN-Fs, RN-Is, and RN-Ds in the configuration. The number of monitors determines the number of unique logical threads that can concurrently access the HN-I system exclusive monitors. These threads can be either processor or device threads, and are designated by a unique combination of SrcID and LPID.

HN-P does not support exclusive accesses.

All exclusives targeting the HN-I are terminated at the HN-I and are not propagated downstream. Exclusives are terminated regardless of the value of the HN-I PoS control register and auxiliary control register.

3.9.2.3 CML support for exclusive accesses

CMN-700 CML supports exclusive accesses in some circumstances. Support for these transactions and the guidance for configuration depends on whether SMP mode is enabled.

CML_SMP mode

In SMP mode, CMN-700 CML supports remote exclusive accesses on an SMP link.

Remote exclusive accesses from an RN-I or RN-D are not supported.

Support for remote exclusive accesses includes these constraints:

- RA in local CCG block passes Excl and LPID fields of incoming CHI request on request message.
- HA in the remote CCG block extracts these bits from request message USER (Ext) field. HA then sends these bits on respective CHI Excl and LPID fields. HA sets the source type as RN-F based on its RAID to LDID register.
- Exclusive OK (EXOK) response is sent

HN-Fs and HN-Is monitor exclusives from remote RN-Fs using existing exclusive monitors. To track remote exclusives, the monitors track the source type, HA_LOGICAL_ID, LDID, and LPID fields of the incoming request.

3.9.2.4 Exclusive accesses in RN-I and RN-D

RN-I supports up to two active exclusive threads at any given AXI port. To differentiate the exclusive threads, RN-Is provides a per port 11-bit mask to extract the bit from AxID.

The 11-bit mask por_{rni,rnd}_s<X>_port_control can be found in the respective RN-I and RN-D AXI port control registers.

3.9.3 Barriers

Barriers were deprecated from CHI-B onwards. All managers, fully coherent and I/O coherent, must handle barriers at the source.

When memory barrier ordering or completion guarantees are required, managers must wait for the responses from all required previous transactions that are issued into the interconnect. No barrier requests can be issued into the interconnect.

All requesting devices that are attached to an interconnect must have a configuration option or strap that prevents issuing of any barriers. If a barrier is issued into the interconnect, the results are **UNPREDICTABLE**.


Note

The DVM_SYNC command, the DVM synchronization that an Arm DSB instruction might initiate, is sent to the DVM block, and executes appropriately.

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

3.9.4 DVM messages

If an RN-F supports *Distributed Virtual Memory* (DVM) messages, it can send DVM requests and receive DVM snoops.

The DVM Node (DN) in CMN is responsible for handling DVM messages. CMN supports up to 4 DNs per chip. DNs are present in HN-D, HN-T, and HN-V devices.

The minimum number of DNs is 1 using HN-D, and the maximum number of HN-D + HN-T + HN-V is 4.

Each DN is part of a DN domain, a DN domain is defined as follows:

- A group of RNs and CXRAs and an associated DN

- The RNs and CXRAs assignment is achieved by configuring XPs to a DN domain, when configuring the mesh.
- A DN domain must be built using contiguous XPs
- A DN domain must contain only one DN: HN-D, HN-T, or HN-V
- All the RN-Fs and CXRAs in a DN domain send DVM messages (OPs and Syncs) to the associated DN

A DVM request from an RN-F is sent to DN in the RN-F's DN domain. On receiving the DVM message, the DN:

- Forwards the DVM message as a snoop to the participating RNs and CXRAs in its DN domain
- Forwards the DVM message to peer DNs in the other DN domains
  - The Peer-DNs snoop the participating RNs and CXRAs in their respective DN domains
  - Collect the Snoop responses
  - Send a single Snoop resopnse to the initiating DN
- Collects the individual snoop responses
- Sends a single response back to the RN-F that originated the DVM message transaction

The default RNs and CXRAs association done while configuring the XP can be reconfigured at boot time using registers:

Each DN has registers to associate the RNs and CML RA within the DN domain (por_dn_domain_rnf0, por_dn_domain_rnf1, por_dn_domain_rnf2, por_dn_domain_rnf3, por_dn_domain_rnd, por_dn_domain_cxra), and each RN and CML RA has a configurable register to associate the DVM Node in its DN domain (dn_nodeid in rnsam_status).


Note

Restrictions for Multi-chip systems: All CCGs communicating (sending and receiving) DVM messages to a given remote chip must be assigned to the same DN domain. Not doing so can result in deadlocks.

The SYSCOREQ/SYSCOACK mechanism provides proxy snoop responses in scenarios when the RN is powered down. For more SYSCOREQ/SYSCOACK information, see 3.3.8 RN entry to and exit from Snoop and DVM domains on page 98.

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. When using an RN-F, only one outstanding DVMOp (Sync) can be issued.

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

3.9.4.1 Support for early completion of DVMOp requests

CMN-700 DN and CML RA nodes can give early completions for DVMOp requests. You enable or disable this mode with programmable register bits.

The following programmable bits enable or disable this mode in these nodes:

DN disable_dvmop_early_comp bit in por_dn_aux_ctl register
CCRA dvm_earlycomp_en bit in por_ccg_ra_aux_ctl register

By default, the early DVMOp completion mode is disabled at HN-D in DNs. When you enable early DVMOp completion, the following errors are not reported as NDE on Comp:

- Poison, DataCheck, and Data Error on RXDATFLIT
- NDE on Snoop responses

CML RA can give early completions for DVMOp requests that are sent over a CML_SMP link. By default, this mode is enabled at CML RA. When this mode is enabled, any NDE on DVMOp completion is dropped.

Optimization for offchip DVMOps can be achieved using config regitsers available in the DN:

- broadcast_dvmop_inner (BDI)
- broadcast_dvmop_outer (BDO).

The DN will utilize these configuration registers and the SnpAttr field on the CHI request channel in order to filter non-Sync DVMOps to remote destinations. DN has detection logic to suppress sending Sync DVMOps if no non-Sync DVMOps have been sent to remote

Table 3-70: DVM message broacasts

BDO	BDI	snpattr	Behavior
0	0	x	Don't send to remote
0	1	x	n/a - reserved, treat as 00
1	0	0	Don't send to remote
1	0	1	Send to remote
1	1	x	Send to remote

3.9.4.2 Optimization of DVMOps and DVMSyncs

The CMN-700 RN-F will broadcast all Inner-Shareable and Outer-Shareable forms on CHI, with the shareability domain indicated on TXREQ.SnpAttr (0=IS, 1=OS). Non-shareable TLBI forms do not cause CHI broadcasts.

All TLBI* instructions exist as non-shareable (non-broadcast), Inner-Shareable, and Outer-Shareable forms.

For example, TLBI ALLE1, TLBI ALLE1IS, TLBI ALLE1OS all represent the same invalidation operation, but at different levels of shareability.

The interconnect *Mesh Network* can use the SnpAttr of a DVMOp request to differentiate local vs. remote chips. Mapping any Inner-Shareable forms (SnpAttr = 0) to the local chip, and thus causing only broadcasts to RN-F and RN-I on the local chip. While mapping Outer-Shareable forms (SnpAttr = 1) to remote chips, causing broadcasts across CXL to remote chips in addition to the local chip. Using this scheme requires that the users OS's at EL1 use TLBI*IS forms and that users do not span across chips. The hypervisor at EL2 would use TLBI*OS and can span across chips.

The *Mesh Network* in the interconnect provides a similar filtering function and applies it to the local chip and remote chips independently based on hardware detection of DVMMsg broadcasts to the two domains since the last DVMSync.

A DVMSync broadcast is caused by the DSB instruction. In general, RN-Fs have the ability to detect whether there have been any TLBI or ICI broadcasts to CHI since the last DSB. If no such broadcasts have occurred, then the DVMSync broadcast is not necessary and thus not sent.

DVMSyncs will only be sent to the Outer Sharable domain if a previous DVMOp has been sent to the Outer Sharable domain. Subsequent DVMSyncs that are received without an Outer Sharable DVMOp preceding them will not be sent to the Outer Sharable domain.

By default, the `BROADCAST_DVMOP_INNER` (BTI) and `BROADCAST_DVMOP_OUTER` (BTO) registers are enabled, indicating that DVMOps and DVMSyncs will broadcast to both Inner and Outer Sharable domains.

The BTI and BTO registers in combination with the `SNPATTR` can be used to prevent sending DVMOps and DVMSyncs to Outer Sharable domains. The DVMSYNC operation does not utilize the SnpAttr field. However, in the BTI=1, BTO=0 configuration, DVMSyncs will not be broadcast to Outer Sharable domains if no previous Outer Sharable DVMOp requests have been received. Subsequent DVMSyncs will be broadcast to Inner Sharable domains only until the next Outer Sharable DVMOp request is received.

The following table is used to determine if a DVMOp will broadcast to the Outer Sharable domain:

Table 3-71: DVM message broacasts

BTO	BTI	SnpAttr	Behavior
0	0	x	Don't send to remote

BTO	BTI	SnpAttr	Behavior
0	1	x	n/a - reserved, treat as 00
1	0	0	Don't send to remote
1	0	1	Send to remote
1	1	x	Send to remote

3.9.4.3 Remote DVM Optimization based on VMID filtering

TheCMN-700 can optimize Outer-Shareable DVMOp requests using VMID filtering.

The DVMOp can be optimized using a VMID filter when:

- The DVM type is TLBI or VICI
- CHI_VMID_VALID = 1

The VMID value is a sequence of the CHI_REQ fields:

- CHI_VMID = {DVM_VMID_Ext, DVM_VMID}

The VMID is compared against all VMID filtering registers and determines all the snoopable targets

```
for(i=0; i<num_vmf_registers; i++)
    if (vmf_reg_valid[i] & ((chi_vmid & vmf_reg_mask[i]) ==
(vmf_reg_vmid[i] & vmf_reg_mask[i])))
        vmid_targets = vmid_targets | vmf_reg_targets[i]
        vmf_match = vmf_match | 1'b1;
    else
        vmid_targets = vmid_targets
        vmf_match = vmf_match
targets = vmf_match ? all_targets & vmid_targets : all_targets;
```

The final DVM target vector will be the combination of all matching vmf_reg_targets. If the VMID value does not match with any target registers, then the DVMOp will target all RNF, RND and CXRA broadcasts.


Note

DVMSyncs are not VMID filterable as the VMID is not valid on CHI

3.9.4.3.1 VMID Filtering Optimization

The DN will broadcast VMID Filterable DVMOPs to all RNFs, RNDs, and CXRAs on a VMID Filter miss.

You can optimize the VMID Filterable DVMOps to prevent the broadcast to the CXRA in the event of a VMID filter miss.

The following configurable bit enables or disables this mode for the CXRA node:

- DISABLE_REMOTE_BROADCAST_ON_VMF_MISS

Enabling the DISABLE_REMOTE_BROADCAST_ON_VMF_MISS configuration bit, will only send the VMID Filterable DVMOp to the RNFs and RNDs in the DN's domain. It will not be broadcast to remote targets.


Remember

DVMSyncs are not VMID Filterable as the VMID filed is not valid on CHI.

3.9.5 Completer Busy indication

Transaction completers can use the *Completer Busy* (CBusy) field to indicate their current level of activity. RNs use this indication to determine whether to throttle outgoing traffic.

CMN-700 implements the CBusy indication function in the following node types:

- HN-F
- SBSX
- CCRA

DN, HN-I, HN-P, and HN-T always drive the CBusy values as `0b000`.


Note

References to SN-F nodes in this section also apply to SBSX

HN-F CBusy

HN-F uses the *Point-of-Coherency Queue* (POCQ) occupancy level to indicate its current activity level. The following table shows the default CBusy values for a 32-entry POCQ. These values represent the default HN-F CBusy response behavior to RNs.

Table 3-72: HN-F POCQ CBusy thresholds for 32e POCQ

CBusy[2:0]	Tracker occupancy level
`0b011`	≥24
`0b010`	≥16
`0b001`	≥8
`0b000`	<8

HN-F supports a multisource mode indication in CBusy responses to the RNs. The CBusy[2] bit is used to represent this mode. A CBusy[2]=1 indicates two or more RNs have

outstanding requests pending in the HN-F POCQ at the time of response. This mode is enabled by default. HN-F can be programmed to exclude any outstanding RN-I requests in the POCQ when calculating multisource mode. It can be enabled by programming the `cmn_hns_cbusy_limit_ctl.hnf_cbusy_mtbit_exclude_rni` to `1'b1`.

HN-F also supports an alternate mode for the CBusy[2] field. It can be programmed to respond with CBusy[2] = 1 when the total number of active requests in the POCQ exceeds a programmable threshold. This mode is in lieu of the multisource mode. The HN-F is programmed to respond busyness in CBusy[1:0] based on read or write request types. The busy indication is only specific to the request category.

The overall occupancy, the total of the read and write request type entries, of POCQ is not available to the RN using CBusy[1:0]. Therefore the above threshold occupancy mode can be used to indicate the complete tracker occupancy and throttle requests accordingly from the RN. Programming cmn_hns_cbusy_mode_ctl.mt_alt_mode_en = 1 enables this mode. The threshold for triggering this above threshold indicator can be programmed in cmn_hns_cbusy_mode_ctl.poc_high_watermark.

SBSX CBusy

SBSXs only drive CBusy on returning TXDAT flits targeting RNs. These nodes use two hierarchical trackers to drive the CBusy field: ReqTracker and DART. The CBusy field reflects the occupancy levels of both trackers combined. Similar to HN-F, the activity thresholds are programmable. The following table shows the default occupancy threshold for 96 entry trackers.

Table 3-73: SBSX tracker CBusy thresholds

CBusy[2:0]	Tracker occupancy level
`0b011`	≥72
`0b010`	≥48
`0b001`	>24
`0b000`	<24

SBSXs do not use the multi-source mode bit, so CBusy[2] is always set to `0b0`.

CML CBusy

RA uses the request tracker (RHT) occupancy level to indicate the current activity level. The following table shows the default values for a 256 entry RHT. This behavior is the default mode of the RA outgoing CBusy in all responses to RNs.

Table 3-74: RA RHT CBusy thresholds

CBusy[2:0]	Tracker occupancy level
`0b011`	≥192
`0b010`	≥128
`0b001`	≥64
`0b000`	<64

RA does not use the multi-source mode bit, so CBusy[2] is always set to `0b0`.

In addition, in SMP mode, CCG implements passing through CBusy from remote HN-F, HN-I, or HN-P to the requesting RN.

CCRA implements the following software configurable bits to send:

1. CCRA CBusy
2. Remote CBusy coming on data or dataless late completion responses. Intermediate responses, such as DBID, CompDBID, and ReadReceipt, have RA CBusy.
3. Greater of the above two: Applicable to cases where remote CBusy can be sent.

See the link [X]_cbusy_prop_ctl bits in 4.3.3.18 por_ccg_ra_ccprtcl_link0_ctl on page 350 register.

3.9.5.1 Advanced CBusy handling in HN-F

CMN-700 HN-F supports advanced CBusy handling and request throttling to SN-F.

HN-F to RN CBusy handling

The responses that are sent from HN-F to RN through RSP and DAT channels carry CBusy values. HN-F has multiple different modes to determine how the CBusy values are specified in the response messages.

HN-F can be configured to respond to RNs with a CBusy value that reflects one of the following options:

- Total number of outstanding requests in the HN-F POCQ (default mode)
- Independent CBusy values for reads and writes:

  CompData type requests (All Read* requests)

  CBusy value is based on number of outstanding reads in the POCQ

  Comp type requests (Writes, Evict, atomics, CMOs)

  CBusy value is based on number outstanding writes in the POCQ
- Return SN-F CBusy value instead of returning value that is based on HN-F POCQ:
  - Read requests receive the Read CBusy of the SN-F
  - Write requests receive the Write CBusy of the SN-F
- Return whichever CBusy value is the highest between HN-F POCQ and SN-F
- MPAM Part ID based CBusy propagation from SN-F

Comp type requests can be further filtered into the following categories:

- CopyBack type requests (Evict, WriteClean, WriteEvictFull, or WriteBack*)
- NonCopyBack type requests (including WriteNoSnp*, WriteUnique*, Combined Write, (P)CMOs, and atomics)

Write filtering of CopyBack versus NonCopyBack types is only supported when you configure HN-F to respond with the CBusy of the POCQ. Write filtering is not supported if the HN-F returns the CBusy value of the SN-F.

The following table shows the format of the cmn_hns_cbusy_limit_ctl_register. This register controls the HN-F CBusy threshold for Read requests.

Table 3-75: cmn_hns_cbusy_limit_ctl register for CBusy thresholds, all requests or read types

Bits	Name	Description
[7:0]	cmn_hns_cbusy_low_limit	Specifies the POC valid threshold at which HN-F is considered least busy
[15:8]	cmn_hns_cbusy_med_limit	Specifies the POC valid threshold at which HN-F is considered medium busy
[23:16]	cmn_hns_cbusy_high_limit	Specifies the POC valid threshold at which HN-F is considered very busy
[48]	cmn_hns_cbusy_rd_wr_types_en	Allows separate CBusy values for reads versus writes. When enabled, the thresholds in this register are only applicable to read type requests. Otherwise these values are the default thresholds for calculating CBusy for all request types in POCQ of the HN-F. This bit must be set when sn_cbusy_prop_en = `0b1` to propagate the SN CBusy.
[63]	cmn_hns_cbusy_mtbit_exclude_rni	Allows HN-F to ignore outstanding read requests from RN-I when calculating busyness

The following table shows the format of the cmn_hns_cbusy_write_limit_ctl register. This register controls the HN-F CBusy threshold for Write requests.

Table 3-76: Register for CBusy thresholds, write requests

Bitfield	Field	Description
[7:0]	cmn_hns_cbusy_low_limit	Specifies the POC valid threshold at which HN-F is considered least busy
[15:8]	cmn_hns_cbusy_med_limit	Specifies the POC valid threshold at HN-F is considered medium busy
[23:16]	cmn_hns_cbusy_high_limit	Specifies the POC valid threshold at which HN-F is considered very busy
[48]	cmn_hns_cbusy_sep_copyback_types	When set, HN-F calculates CBusy based on outstanding CopyBack and NonCopyBack type requests independently in the HN-F POCQ

The following table shows the CBusy values that are returned to RNs according to programming.

Table 3-77: HN-F CBusy value propagation according to programming

cmn_hns_adv_cbusy_m	cmn_hns_cbusy_rd_wr_ty	sn_cbusy_prop_en	cbusy_highest_of_all_e	cbusy_mpam_tbl_er	CBusy value passed to RN
`0b0`	x	x	x	`0b0`	POCQ CBusy value is returned
`0b1`	`0b0`	x	x	`0b0`	POCQ CBusy value is returned
`0b1`	`0b1`	`0b0`	`0b0`	`0b0`	POCQ CBusy value for read or write is returned, according to the request type
`0b1`	`0b1`	`0b1`	`0b0`	`0b0`	SN CBusy value for read or write is returned for the corresponding SN group, according to the request type

cmn_hns_adv_cbusy_m	cmn_hns_cbusy_rd_wr_ty	sn_cbusy_prop_en	cbusy_highest_of_all_e	cbusy_mpam_tbl_er	CBusy value passed to RN
0b1	0b1	x	0b1	0b0	Highest of either the SN or POCQ CBusy value for read or write is returned, according to the request type
0b1	0b0	x	x	0b1	Reflects the SN CBusy for the corresponding MPAM PartID in both Read and Write response
0b1	0b1	x	x	0b1	Reflects MPAM partID based SN CBusy for Write responses. Read responses receive POCQ occupancy based CBUSY in RSP/DATA flits.

Where applicable, HN-F returns the read or write CBusy value according to opcode type.

Write CBusy values can be further separated into CopyBack and NonCopyBack values using the cmn_hns_cbusy_sep_copyback_types field. This separation only applies when HN-F is programmed to propagate the POCQ CBusy values.

In this mode, CopyBack write type values account for pending WriteClean*, WriteBack*, WriteEvictFull*, and Evict type operations.

NonCopyBack write type values account for all other pending write operations (WriteUnique*, WriteNoSnp*).Combined Write* and (P)CMO operations are counted towards NonCopyBack types. Standalone CMOs are not counted towards either of the CopyBack or NonCopyBack type requests.

HN-F to SN-F CBusy based throttling

HN-F can identify two groups of memory controllers using a configuration bit for each SN. These groups are known as Group A or Group B. You can use the two groups to identify fast and slow memory types. Therefore, the HN-F can handle traffic to and from the two types independently of each other.

HN-F can track the read and write busyness to each SN-F group over a configurable transaction window. It can be programmed to track the last 128 or 256 transactions. When HN-F has received as many responses from SN-F, it measures the current busyness for each group of SN and request types (read and write). The measured busyness is then used to throttle the traffic to SN-F appropriately.

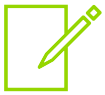
The threshold for measuring the CBusy for the last 128 or 256 transactions is also configurable. For example, consider a scenario where HN-F is programmed to calculate the last 128 CBusy

responses. HN-F tracks the number of times it receives CBusy values of `0b00`, `0b01`, `0b10`, and `0b11` for each SN group.

In this example, the HN-F receives more than 16 CBusy = `0b11` responses from Group 0 SN-Fs out of the last 128 responses. In this case, HN-F treats the final SN-F CBusy value as `0b11` for the subsequent 128 transactions while continuing to accumulate new CBusy response values.

HN-F can be configured to throttle outgoing requests in either a static mode or a default dynamic mode:

- Static throttling mode: HN-F controls the outstanding transactions at any point for a given SN group and request to values as programmed in cmn_hns_cbusy_resp_ctl register:
  - CBusy = `0b11` (Very busy): HN-F will throttle back outstanding transactions to value programmed in cbusy_sn_static_ot_count_cbusy11 field.
  - CBusy = `0b10` (Medium busy): HN-F will throttle back outstanding transactions to value programmed in cbusy_sn_static_ot_count_cbusy10 field.
  - CBusy = `0b01` (Low busy): HN-F will throttle back outstanding transactions to value programmed in cbusy_sn_static_ot_count_cbusy01 field.
  - CBusy = `0b00` (Not busy): HN-F can issue as many requests as the number of POCQ entries
- Dynamic throttling mode: The number of *Outstanding Transactions* (OTs) can be dynamically throttled according to programmed values. It can be configured to increment or decrement the transaction count by one, two, four, or eight transactions after every 128 or 256 transaction window (as programmed).
  - CBusy = `0b11` (Very busy): Decrement the OT count
  - CBusy = `0b10` (Medium busy): No change to the current OT count
  - CBusy = `0b01` (Low busy): Increment the OT count
  - CBusy = `0b00` (Not busy): Increment the OT count


Note

cbusy_sn_static_ot_count_cbusyXX fields must never be programmed to 0 as it can stall forward progress.

Throttling request to SN can be disabled by programming cbusy_sn_req_throttle_dis config bit to `0b1`. When you configure an HN-F to respond to RNs with the CBusy value of an SN-F, HN-F can propagate the CBusy value according to the SN-F group that the request targets. For example, consider an RN-F sending a read request that is targeting SN group A. The RN can receive the CBusy value for a group A SN, even if the request hits in SLC and therefore the HN-F completes the request.

The following table shows the format of the cmn_hns_cbusy_resp_ctl register. This register controls the CBusy responses.

Table 3-78: cmn_hns_cbusy_resp_ctl register for configuring CBusy value on responses

Bits	Name	Description: Controls the CBusy responses
[0]	sn_cbusy_prop_en	When set to `0b1`, HN-F responds with the CBusy values from SN-F instead of using its own POCQ occupancy-based thresholding. Read and write modes are still controlled using cmn_hns_cbusy_limit_ctl and cmn_hns_cbusy_write_limit_ctl.
[4]	Cbusy_highest_of_all_en	When set to `0b1`, HN-F responds with the CBusy values from the highest of group A and group B
[7]	cbusy_sn_static_ot_mode_en	Enables the static OT throttling to SN
[21:16]	cbusy_sn_dynamic_ot_count	Count by which the OT count is incremented or decremented for dynamic OT throttling

HN-F throttles only dynamic credit requests to SN-F by default. Requests that were retried and have received a static credit grant from SN-F are allowed to bypass the throttling mechanism. HN-F can be programmed to also throttle static credit requests by setting the cbusy_sn_retried_req_throttle_en field to `0b1`.

The following table shows the format of the cmn_hns_sam_sn_properties1 register. This register controls the group to which each SN belongs.

Table 3-79: Per SN group identifier in cmn_hns_sam_sn_properties registers

Field	Description
sn0_group	**0b0** Group A **0b1** Group B
sn1_group	
sn2_group	
sn3_group	
sn4_group	
sn5_group	
sn6_group	
sn7_group	
Region0_sn_group	
Region1_sn_group	

The following table shows the format of the cmn_hns_cbusy_sn_ctl register. This register controls the CBusy sampling.

Table 3-80: cmn_hns_cbusy_sn_ctl register for CBusy sampling control

Bitfield	Field	Description
[9:0]	cmn_hns_cbusy_threshold_cntr01	CBusy threshold at which SN-F is considered busy for Counter_01
[25:16]	cmn_hns_cbusy_threshold_cntr10	CBusy threshold at which SN-F is considered busy for Counter_10
[41:32]	cmn_hns_cbusy_threshold_cntr11	CBusy threshold at which SN-F is considered busy for Counter_11
[56:48]	cmn_hns_cbusy_txn_cnt	Number of SN responses over which the CBusy counters are tracked

HN-F continues to propagate the multi-source bit (CBusy[2]) in the advanced modes.

HN-F supports MPAM PartID based CBusy propagation to RN-F. This mode is enabled by programming mpam_tbl_en=1. When enabled, HN-F captures the last CBusy value in Read and

Write responses from SN for each MPAM PartID. This CBusy value is then propagated to RN-F for corresponding MPAM PartID responses.

For specific use cases where the SN CBusy value must be aggregated in the MPAM Part ID table, HN-F supports an alternate mode of CBusy capture for SN responses. When this mode is enabled using cmn_hns_cbusy_mode_ctl.cbusy_alt_mode_en=1, HN-F compacts the 3 bits of SN CBusy into 2-bit CBusy as follows:

- Default_Mode: CBusy[1:0] = SN_CBusy[1:0]
- Alt_Mode: CBusy[1] = SN's CBusy[2], CBusy[0] = (SN_CBusy[1] & SN_CBusy[0])

HN-F also supports an alternate mode of CBusy[2] to indicate the POCQ occupancy being higher than a certain threshold. This mode is enabled when mt_alt_mode_en=1. The following table shows the mt_alt_mode modes. For example, if mt_alt_mode_en=0, CBusy[2]=1 indicates that there are requests from more than one RN active in the POCQ.

Table 3-81: mt_alt_mode modes

Mode	Mode	Description
mt_alt_mode_en	cmn_hns_cbusy_rd_wr_types_en	CBusy[2] MT bit
`1'b0`	X	Multi source mode: Requests from more than one RN is active in POCQ
`1'b1`	`1'b0`	POC occupancy is higher than poc_high_watermark
`1'b1`	`1'b1`	Write responses: POC occupancy > poc_high_watermark , Read responses: Requests from more than one RN is active in POCQ

3.9.6 REQ RSVDC propagation

CMN-700 supports the propagation of the *Reserved for customer use* (RSVDC) field of the CHI REQ flit through the interconnect.

Calculate the width of the RSVDC field that is propragated throughout the mesh using configuration parameters:

`REQ_RSVDC_WIDTH` + (`RSVDC_PBHA_MODE_EN` X `RSVDC_PBHA_WIDTH`) + `RSVDC_STRONGNC_EN`

For a multi-chip system, the REQ RSVDC field is preserved and only passed over a CML_SMP link. For AXI subordinate interfaces, the incoming AxUSER field is mapped to CHI REQ RSVDC field and is propagated through the interconnect. For AXI manager interfaces, CHI REQ RSVDC field is mapped to AxUSER. This field is not stored in SLC and so is not preserved for requests that are allocated in SLC.

3.9.7 DAT RSVDC propagation

CMN-700 supports propagation of the RSVDC field of the CHI DAT flit through the datapath and SLC for full cache line read and write operations from RN-Fs to SN-Fs. This support includes CML traffic when in CML_SMP mode.

Calculate the width of RSVDC field that is propragated throughout the Mesh as follows:

4 + (RSVDC_METADATA_MODE_EN * 12) + (RSVDC_PBHA_MODE_EN X RSVDC_PBHA_WIDTH) + RSVDC_STRONGNC_EN

This feature is enabled by setting the RSVDC_METADATA_MODE_EN parameter to 1. For requests that are allocated in SLC, DAT.RSVDC is also stored in the SLC.

The RSVDC is not preserved for AXI traffic that RN-I or RN-D initiate, or targeting SBSX, HN-I, HN-T, HN-P, or HN-D. It is also not preserved for non-SMP CML links.

The RUSER and WUSER signal widths increase on ACE-Lite manager and subordinate interfaces when RSVDC_METADATA_MODE_EN is set to 1. However they are not used to propagate DAT RSVDC values.


Note

Partial cache states are not supported when the RSVDC_METADATA_MODE_EN parameter is set.

WriteNoSnpZero and WriteUniqueZero opcodes do not update DAT.RSVDC value.

DAT.RSVDC propagation and storage in SLC is only avaliable if CHI_MTE_ENABLE is 0.

3.9.8 PBHA RSVDC handling

CMN-700 supports an optional *Page Based Hardware Attributes*PBHA subfield on REQ.RSVDC and DAT.RSVDC field of the CHI REQ and DAT flit respectively.

The PBHA subfield and its associated functionality can be enabled by setting the RSVDC_PBHA_MODE_EN parameter to 1. Number of PBHA bits can be controlled using the RSVDC_PBHA_WIDTH parameter.

PBHA subfield on the RSVDC field is propagated through RN-F, RN-I, RN-D, HN-F, and CML_SMP blocks.


Note

The RN-I and RN-D use AxUSER to send PBHA value in CMN-700.

A PBHA subfield is not preserved for AXI traffic targeting SBSX, HN-I, HN-T, HN-P, or HN-D. it is also not preserved for non-SMP CML traffic.

PBHA value is transferred back to RN-F on DAT.RSVDC upon a read. RN-I,RN-D do not propagate PBHA field on xUSER. RN-F also transfers PBHA value to HN-F on DAT.RSVDC with Snoop data. SN-F bound request inherits the PBHA value populated from original requestor or from SLC. SN-F do not send the PBHA value back to HN-F or RN-F and the return PBHA value from SN-F on DAT.RSVDC is expected to be 0 in those cases.

3.9.9 StrongNC RSVDC handling

CMN-700 supports an optional StrongNC subfield in the RSVDC field on CHI REQ channel and AxUser field on AXI interface.

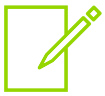
To enable the StrongNC subfield and its associated functionality, set the `RSVDC_STRONGNC_EN` parameter to 1.

StrongNC subfield in the RSVDC field is propagated through RN-I, RN-D, HN-F, HN-I, HN-P, HN-D, HN-T, and CML_SMP blocks. StrongNC subfield is not preserved for traffic that targets SBSX or non-SMP CML links.

For StrongNC requests, HN-F propagates StrongNC subfield in REQ.RSVDC field and sends StrongNC requests directly to SN-F. HN-F does not return data from SLC with SLC hit, nor send out snoops with SF hit. SN-F must be able to process StrongNC requests and return data. StrongNC writes clear exclusive monitors inside HN-F.


Note

RN-F does not support StrongNC requests and always drive the StrongNC bit in the REQ.RSVDC field to 0. RN-I and RN-D support non-cacheable, non-exclusive ReadNoSnp, WriteNoSnpFull, WriteNoSnpPtl with INVALID TagOp for StrongNC requests.


Note

If StrongNC requests fall into an HN-F OCM region, HN-F ignores the StrongNC subfield and process the requests like normal OCM requests.

3.9.10 REQ and DAT RSVDC for CML non-SMP links

If a CCG block is used to connect to a CXL device (i.e. non-SMP link), then RSVDC fields of REQ and DAT flits can be driven from the contents of REQ_RSVDC_OVRD_VAL and DAT_RSVDC_OVRD_VAL straps. This is enabled by:

1. lnk<X>_req_rsvdc_ovrd_en and lnk0_dat_rsvdc_ovrd_en bits in HA's por_ccg_ha_cxprtcl_link<X>_ctl register
2. Setting lnk<X>_dat_rsvdc_ovrd_en bits in RA's por_ccg_ra_ccprtcl_link<X>_ctl register

3.10 Processor events

CMN-700 supports communicating processor events to all processors in the system.

See the processor event interface signals that B.15 Processor event interface signals on page 1301 describes.

When a processor generates an output event that an SEV instruction triggers, it is broadcast to all processors in the system. Similarly, anytime an exclusive monitor within HN-F or HN-I is cleared, an output event is broadcast to all processors in the system. The event interface signals are also present at RN-I and RN-D nodes, for use by components such as a *System Memory Management Unit* (SMMU).

The logical operator OR is used to combine the EVENT signals, then the result is broadcast to the processors.

3.11 Quality of Service

CMN-700 includes end-to-end QoS capabilities which support latency and bandwidth requirements for different types of devices.

The QoS device classes are:

Devices with bounded latency requirements

These devices are primarily real-time or isochronous that require some or all of their transactions 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

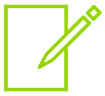
The performance of these devices is highly impacted by the response latency that is incurred by their transactions. Processors are traditionally highly latency-sensitive devices, although a processor can also be a bandwidth-sensitive device depending on its workload.

Bandwidth-sensitive devices

These devices 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 devices 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 can be classified into one or more of these classes, depending on its workload requirements.

Support for these different types of devices and their resulting traffic is included in the AMBA® 5 CHI protocol and in the entirety of CMN-700 microarchitecture. Each component in CMN-700 contributes to the overall QoS microarchitecture.

3.11.1 Architectural QoS support

The AMBA 5 CHI protocol includes a 4-bit *QoS Priority Value* (QPV) with all message flits.

The QPV of the originating message must propagate for all messages in a transaction. The QPV is defined as higher values being higher priority and lower values being lower priority. All CMN-700 components use the QPV to provide prioritized arbitration and to prevent Head-of-line blocking based on the QPV.

3.11.2 Microarchitectural QoS support

The QPV of RN requests must be modulated depending on how well or poorly their respective QoS requirements are met.

3.11.2.1 QoS regulators

Although the QoS-modulation capability can be integrated into the RN, CMN-700 enables system designers to include non-QoS-aware devices in the CMN-700 system, but still have these devices meet the QoS-modulation requirements of the CMN-700 QoS microarchitecture.

CMN-700 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 CMN-700:

- For CHI ports, the regulator is present in the XP
- For ACE-Lite/AXI4 subordinate interfaces, the regulator is present at the ACE-Lite/AXI4 side of the protocol bridge

CMN-700 QoS regulators have three operating modes:

- Pass-through
- Programmed QoS value
- Regulation

These operating modes are controlled through memory-mapped configuration registers.

3.11.2.2 QoS regulator operation

The values of the base QPV, AxQOS for AXI and ACE-Lite interfaces or RXREQFLIT.QOS for CHI ports, are inputs to the QoS subblock.

When latency regulation or period regulation is enabled, the values generated by the regulators replace the base QPV values. For an RN-F, a single QoS regulator monitors CHI transactions that

return data to the RN-F such as reads, atomics, and snoop stash responses. The regulated QPV is applied to all CHI requests from that RN-F. For an RN-I or RN-D, 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, RN-D, and XP.

Latency regulation mode

When configured for latency regulation, the QoS regulator increases the QPV whenever actual transaction 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 increases by a fractional amount. The scale factor K_i determines this amount.
- For every cycle that the latency of a transaction is less than the target latency, the QPV decreases by a fractional amount. The scale factor K_i determines this amount.

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

The 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^{-3} and a programmed value of `0x7` = 2^{-10}.

The QoS regulator can be programmed to operate in latency regulation mode by programming the following bits in the QoS Control register:

- Set the qos_override_en bit to `0b1`.
- Set the lat_en bit to `0b1`.
- Set the reg_mode bit to `0b0`.
- Set the pqv_mode bit to `0b0`.

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 (as measured at dispatch time) is more than the target period, the QPV increases the scale factor K_i by a fractional amount.
- For every cycle that the period between transactions is less than the target period, the QPV decreases the scale factor K_i by the same fractional amount.

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

The 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^{-3} and a programmed value of `0x7` = 2^{-10}.

The QoS regulator can be programmed to operate in period regulation mode by programming the following bits in the QoS Control register:

- Set the qos_override_en bit to `0b1`.
- Set the lat_en bit to `0b1`.

- Set the reg_mode bit to `0b1`.

There are two modes of period regulation:

Normal mode

The QPV does not increase or decrease when there are zero outstanding transactions.

Quiesce high mode

The QPV increases by a fractional amount, which the scale factor K_i determines, in every cycle where there are zero outstanding transactions.

The mode of period regulation can be selected by programming the pqv_mode bit in the QoS Control register.

3.11.2.3 RN-I and RN-D bridge QoS support

In addition to the QoS regulators, the RN-I and RN-D bridge provides QoS-aware arbitration mechanisms.

To simplify arbitration logic, all transactions are split into two QoS Priority Classes (QPCs), high and low. QoS-15 transactions make up the high class. All other transactions are considered to be in the low class.

Port multiplexer arbitration

An RN-I and RN-D bridge includes three ACE-Lite and ACE-Lite-with-DVM ports. The RN-I and RN-D bridge selects between these ports for allocation into its transaction tracker. This selection process makes the allocated transaction a candidate for issuing to a Home Node. The port multiplexer is arbitrated using the following strategy:

- High QPC first, then the low QPC
- Round-robin arbitration among the AMBA ports within a QPC

Tracker allocation

When transactions are allocated into the tracker, they are scheduled for issuance to a Home Node based on QPC. This strategy is the same strategy as port mux arbitration.

- High QPC first, then the low QPC
- Round-robin arbitration in a QPC among the transactions for issue

3.11.2.4 HN-F QoS support

The HN-F includes the following QoS support mechanisms, if configured to use QoS based classes:

QoS decoding in HN-F

The HN-F interprets the 4-bit QPV at a coarser granularity, as the following table shows. See section 5.5 HN-F class-based resource allocation and arbitration on page 1192.


Note

The following table has default configuration and is software-programmable.

Table 3-82: QoS classes in HN-F

QoS value	Class	Dedicated	Contended min	Max allowed
15	Class 0	0	POCQ_ENT / 4	POCQ_ENT - 1
14-12	Class 1	0	POCQ_ENT / 4	POCQ_ENT - 2
11-8	Class 2	0	POCQ_ENT / 4	POCQ_ENT / 2
7-0	Class 3	0	POCQ_ENT / 4	POCQ_ENT / 8

QoS class and POCQ resource availability

The HN-F includes a multi-entry 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. POCQ is partitioned so that different classes can use configurable number of entries as dedicated, max_allowed or contended_min as shown in figure earlier, ensuring bandwidth and latency requirements of higher priority transactions are met. See 5.5 HN-F class-based resource allocation and arbitration on page 1192.

3.11.2.5 HN-I and SBSX QoS support

The HN-I bridge provides QoS-aware arbitration mechanisms for static grants and AMBA requests.

To simplify arbitration logic, all transactions are split into two QPCs: high and low. QoS-15 transactions make up the high class. All other transactions are considered to be in the low class.


Note

SBSX QoS support is identical to that of the HN-I.

Dynamic credit tracker allocation

Requests allocate into the tracker until it is full, after which requests are then retried and the HN-I increments a credit counter for the affected RNs in an internal retry bank.

When a tracker entry is cleared, the HN-I checks the retry bank for any retried transactions. If any are present, the newly available tracker entry is reserved and a static credit grant is sent to an RN chosen using the following algorithm:

- Choose an RN marked as high QPC first, then the low QPC.
- Use round-robin arbitration within a QPC.

Scheduling to AMBA interface

When transactions are allocated into the tracker, they are scheduled for issue on the AMBA interface based on QPC following a similar strategy to static credit allocation:

- Choose high QPC first, then the low QPC.
- Use round-robin arbitration within a QPC.

Write data buffers are also allocated based on QPC class. For write requests that are ready to issue:

- Choose high QPC first, then the low QPC.
- Use round-robin arbitration within a QPC.

3.11.2.6 CML QoS overrides

You can program CMN-700 to override the QPV on incoming transactions through the HA.

You can program the HA config control register, 4.3.2.4 por_ccg_ha_cfg_ctl on page 291, to hold a QoS override value. The HA overrides the QPV value on the CHI side with the value that is programmed into this register.

3.11.3 QoS configuration example

This example configuration demonstrates the QoS mechanisms and their contribution to the overall QoS solution.

It is the responsibility of the SoC designer and system programmer to configure CMN-700 to meet the specific requirements of the system and expected workloads.

System operating conditions

The example QoS configuration assumes the following:

- Four processor clusters:
  - Bimodal operation. A processor cluster is latency-sensitive when bandwidth per cluster is ≤ 2GB/s, and bandwidth-hungry, and therefore latency-tolerant, when bandwidth per cluster is > 2GB/s.
  - 16 outstanding combined reads and writes
  - 10GB/s maximum bandwidth per cluster
  - 25GB/s maximum aggregate bandwidth across all processor clusters
- Four peripheral devices with bounded latency requirements:
  - Each device is the sole device that is connected to ACE-Lite interface 0 on four different RN-I bridges
  - 1 microsecond maximum latency requirement
  - 4GB/s maximum bandwidth per device
  - 210GB/s maximum aggregate bandwidth across all devices

- 14 peripheral bandwidth-hungry devices:
  - Connected to all remaining RN-I ACE-Lite interfaces
  - 12GB/s read or write bandwidth per device, with a combined maximum of 24GB/s
  - 60GB/s maximum aggregate bandwidth across all devices
- All devices can be concurrently active
- 80GB/s maximum aggregate bandwidth across all devices

HN-F QoS classes

For the QoS ranges and class values in HN-F, see 3.11.2.4 HN-F QoS support on page 248.

QoS regulator settings

To meet the bandwidth and latency requirements of the described system configuration, CMN-700 QoS regulators can be configured with the settings that the following table shows

Table 3-83: QoS regulation settings

Device	Regulation type	Regulation parameter	QoS range	QoS scale
Processor	Latency	60ns max latency	11-13	8-9
Real-time peripheral	Override (constant value)	Constant	15	n/a
High-bandwidth peripheral	Override (constant value)	Constant	8	n/a

The latency specification for real-time peripherals must be sufficiently far below the maximum real-time constraint to allow the control loop in the QoS regulator to adjust based on achieved latency, without violating the maximum latency requirement.

To meet the bandwidth and latency requirements of the described system configuration, HN-F QoS reservation values can be configured (based on 32-entry POCQ with one entry for SF back invalidations) as the following table shows and summarizes.

Table 3-84: QoS class and reservation value settings

QoS class	QoS reservation value
HighHigh	31
High	30
Medium	15
Low	5

These settings enable the following system functionality:

- Real-time devices are QPV-15, ensuring their transactions meet their bounded latency requirements
- The processor QPV is higher than the bandwidth-hungry devices, second only to the real-time devices, and therefore generally achieves minimum latency, except in the event of high-bandwidth real-time traffic
- Real-time devices can be configured to have all the HN-F POCQ buffering available to them, to prevent bandwidth limitations from impacting achieved latency

- Real-time devices always have buffering available to them throughout the entirety of CMN-700 preventing Head-of-line blocking from lower-priority or higher-latency transactions

4. Programmers model

This chapter describes the application-level registers and provides an overview for programming the CMN-700 interconnect.

4.1 About the programmers model

A CMN-700 interconnect consists of various components, such as XP, RN-I, or DTC, that are accessed through memory mapped registers for configuration, topology, and status information.

The memory mapped registers are organized in a series of 64KB regions. They are accessed through CHI, AXI, or APB read and write commands. APB accesses to the registers occur through the CMN-700 HN-D APB interface.

A full description of a CMN-700 interconnect consists of a list of components, the compile-time configuration options for each component, and the connectivity between the components. Software can determine the full configuration of the CMN-700 interconnect through a sequence of accesses to the configuration register space.

4.1.1 Node configuration register address mapping

All CMN-700 configuration registers are mapped to a specific address range that is divided into sections for individual components.

The configuration register address space starts at PERIPHBASE. For a system with X and Y dimensions of eight or less, the address space has a maximum size of 256MB. For a system with X or Y dimensions of nine or more, the address space has a maximum size of 1GB.

The CFGM_PERIPHBASE input signal controls the reset value of PERIPHBASE. Configuration register accesses through the HN-D APB interface use the same addressing scheme as the CHI and AXI interfaces. However, only 32 bits of the address are provided to the APB interface.

All configuration, information, and status registers in a CMN-700 interconnect are grouped into 64KB regions each associated with a CMN-700 component instance. The base address of each region can be determined at compile time, or determined at runtime through a software discovery mechanism.

Software discovery consists of three steps:

1. Read information in the 64KB region at offset `0x0`. This information determines the number of XPs in CMN-700 and the offset from PERIPHBASE for the 64KB region of each XP.
2. Read information in the 64KB region that is associated with each XP. This information determines the components that are associated with that XP, topology information for those components, and the offset from PERIPHBASE for each component 64KB region.

3. Read information in the 64KB region that is associated with the component. This information determines the type of block and the configuration details of the component.

For more information about these steps, see 3.5.4 Discovery tree structure on page 201.

With this sequence, software can build a list of all components in the system and the addresses of their respective 64KB configuration regions.

4.1.2 Global configuration register region

The 64KB block at offset `0x0` + PERIPHBASE contains global information and configuration for CMN-700, and the first level of discovery information for components in the system.

Each XP Base Address register contains the offset from PERIPHBASE for a 64KB region that contains the information about one XP. The XP Base Address register also contains discovery information for components that are associated with that XP. The XP Base address refers to the relative address of the XP configuration registers. The first level of Discovery points to each por_mxp_node_info register of the XPs.

For more register information, see 4.3.5 Configuration manager register descriptions on page 446.

4.1.3 XP configuration register region

Each XP has a 64KB configuration register region with information about that XP and all associated components.

See 4.3.13 MXP register descriptions on page 880.

4.1.4 Component configuration register region

Each non-XP component has a 64KB configuration register region. This region has programmable information, status, and configuration options for that component.

The contents are listed in the following table, including the number of 8B registers which fit in the space.

Table 4-1: Configuration register region values

Register sections	Relative offset	Absolute offset	Description
Discovery register section			
NODE INFO (node type, node ID)	`0x0`	`0x0`	Up to 16 registers
CHILD INFO (number of children, offset of the first child pointer register = `0x100`)	`0x80`	`0x80`	Up to 16 registers
CHILD POINTER registers	`0x100`	`0x100`	Up to 256 registers

Register sections	Relative offset	Absolute offset	Description
UNIT REGISTER section	0x900	Unit-specific registers	
UNIT INFO	0x0	0x900	Up to 16 registers
UNIT SECURITY	0x80	0x980	Up to 16 registers
UNIT CTRL	0x100	0xA00	Up to 16 registers
UNIT QoS	0x180	0xA80	Up to 32 registers
UNIT DEBUG	0x280	0xB80	Up to 16 registers
UNIT OTHER	0x300	0xC00	Up to 128 registers
UNIT POWER	0x700	0x1000	4KB-aligned space – 512 registers
UNIT PMU	0x1700	0x2000	4KB-aligned space – 512 registers
UNIT RAS (Secure RAS registers)	0x2700	0x3000	4KB-aligned space – 512 registers
UNIT RAS (Non-secure RAS registers)	0x2800	0x3100	4KB-aligned space – 512 registers

4.1.5 Requirements of configuration register reads and writes

Reads and writes to the CMN-700 configuration registers must meet certain requirements.

If the following requirements are not met, **UNPREDICTABLE** behavior can occur:

- 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® AXI or ACE-Lite interface.
  - BE must indicate that all byte lanes are valid if the write transaction is sent from an AMBA® 5 CHI interface.
- Writes to read-only registers/fields are ignored. Reads to write-only registers read-as-zero. The values in W1C registers can also be read
- Secure registers can only be accessed through a Secure access, that is, NS=0.

    - If a secure register is accessed with a non-secure access it will be treated read-as-zero or write-ignored.
- Non-secure registers can be accessed through either a Secure or Non-secure access.


Note

The various secure register groups override register can change the access restrictions of other registers.

For more information about error signal handling, see 3.8 Reliability, Availability, and Serviceability on page 208.


Note

PSLVERR will not report non-existent XP error or non-existent CFG subordinate on XP where HN-D is located; whereas CHI and AXI will report NDE for such cases.

4.1.6 APB-only access

A dedicated 64K block at offset `1` * 64K + PERIPHBASE is only accessible through the APB interface.

A register por_apb_only_access has 3 control bits that limit access to CMN control registers and DSU/DMC peripheral space.

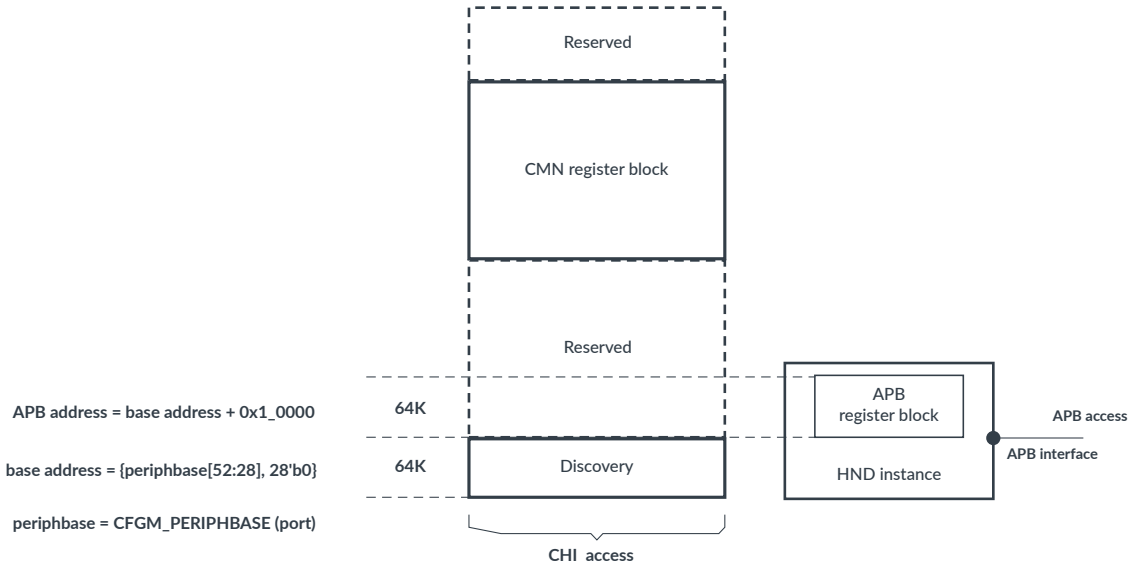
- If cmn_apb_only is set to 1, then all config registers can only be accessed by APB. The write from CHI and AXI is acknowledged and dropped, and read is returned 0.
- If dsu_apb_only is set to 1, then all config registers can only be accessed by APB. The CHI write to DSU space is acknowledged and dropped, and read is returned 0.
- If dmc_apb_only is set to 1, then all config registers can only be accessed by APB. The CHI write to DMC space is acknowledged and dropped, and read is returned 0.

The default value of cmn_apb_only, dsu_apb_only, and dmc_apb_only is 0. And when set at 0, the register access from CHI and/or AXI follows the current config access behavior.

A dedicated APB subordinate port is provided for the access of all CMN-700 configuration registers. The APB subordinate port has the following properties:

- APB only supports 32-bit accesses.
- PSTRB[3:0] must be driven to `4'hF` for a write transaction.
- Secure access requires setting PPROT[1] to 0.

Figure 4-1: APB system memory map access



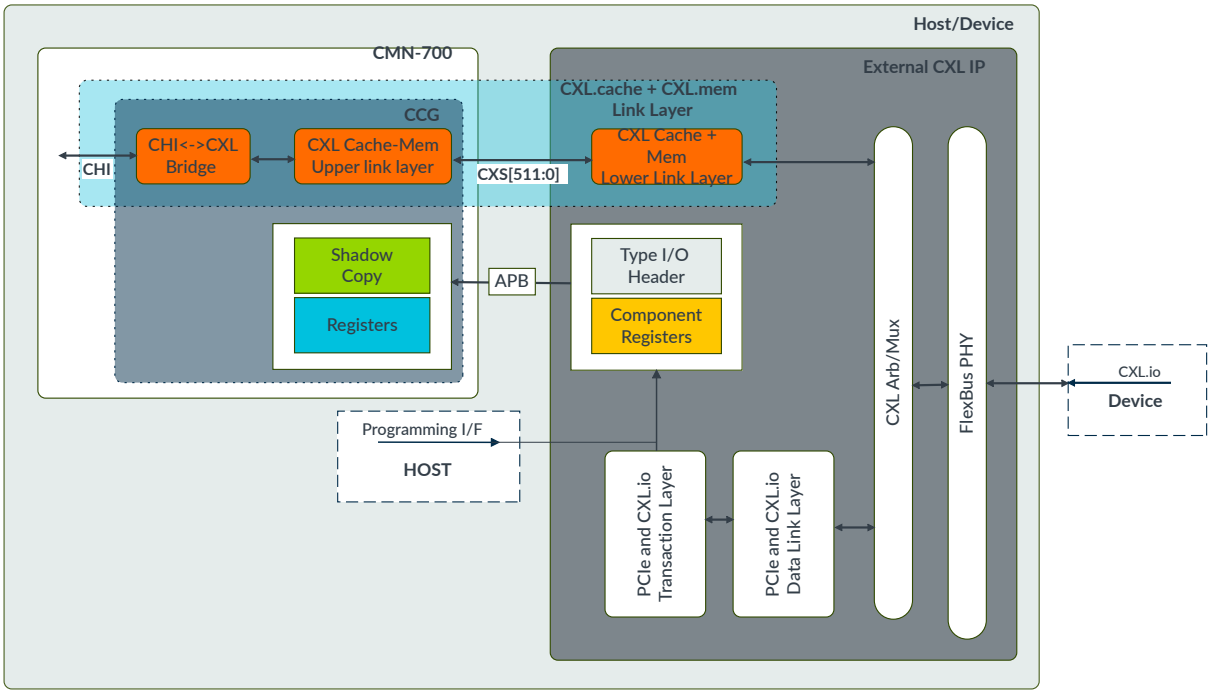
4.1.7 CXL registers

This section covers details about CXL register distribution and how they can be accessed if implemented in CCG.

CXL defined capability structures, which are mapped to CXL Root Port (RP) or CXL Device, can belong in either CXL defined Configuration Space or Memory Mapped Space (Component Registers). The functionality controlled by these capability structures can be implemented entirely in either external CXL controller IP or CML gateway (CCG) or could be distributed across both.

As shown in figure below, an APB port is defined between CXL controller IP and CCG, for CXL controller IP to access registers implemented in each of gateway block.

Figure 4-2: CXL register access



The registers implemented in each of CML gateway blocks (i.e., CCGs) belong to two categories

- Registers which are fully defined in CCG
- Shadow copy of control registers, which are also implemented in CXL controller IP
  - Any write to these registers must update both copies (main and shadow)
  - Any read to these registers must read from the main copy. Shadow copy read is not needed.

For more information about the signal description of the APB port, see B.3 APB interface signals on page 1285.

14-bit APB address (PADDR [13:0]) bus is provided and can access 16K of register space per CCG. CXL registers implemented in CCG are present in pre-defined offset within this 16K space. The based address of each of this 16K space is 0x0.

All the remaining CXL defined registers, which are not defined in CCG, must be implemented in CXL controller IP or SoC


Note

When in CXL1.1 mode, all the registers are considered as memory mapped registers, but this does not have any impact on how these registers are mapped in APB port register address space

4.2 Register summary

The register summary tables list the registers in CMN-700.

4.2.1 APB register summary

This section lists the APB registers used in CMN-700.

APB register summary

The following table shows the *APB* registers in offset order from the base memory address

Table 4-2: APB register summary

Offset	Name	Type	Description
16'h0	por_apb_node_info	RO	4.3.1.1 por_apb_node_info on page 284
16'h80	por_apb_child_info	RO	4.3.1.2 por_apb_child_info on page 285
16'h980	por_apb_only_access	RW	4.3.1.3 por_apb_only_access on page 286
16'h988	por-axu-control	RW	4.3.1.4 por_axu_control on page 287

4.2.2 CCG_HA register summary

This section lists the CCG_HA registers used in CMN-700.

CCG_HA register summary

The following table shows the *CCG_HA* registers in offset order from the base memory address

Table 4-3: CCG_HA register summary

Offset	Name	Type	Description
16'h0	por_ccg_ha_node_info	RO	4.3.2.1 por_ccg_ha_node_info on page 289
16'h8	por_ccg_ha_id	RW	4.3.2.2 por_ccg_ha_id on page 289
16'h80	por_ccg_ha_child_info	RO	4.3.2.3 por_ccg_ha_child_info on page 290
16'hA00	por_ccg_ha_cfg_ctl	RW	4.3.2.4 por_ccg_ha_cfg_ctl on page 291
16'hA08	por_ccg_ha_aux_ctl	RW	4.3.2.5 por_ccg_ha_aux_ctl on page 292
16'hA10	por_ccg_ha_mpam_control_link0	RW	4.3.2.6 por_ccg_ha_mpam_control_link0 on page 294
16'hA18	por_ccg_ha_mpam_control_link1	RW	4.3.2.7 por_ccg_ha_mpam_control_link1 on page 295
16'hA20	por_ccg_ha_mpam_control_link2	RW	4.3.2.8 por_ccg_ha_mpam_control_link2 on page 297

Offset	Name	Type	Description
16'h980	por_ccg_ha_secure_register_groups_override	RW	4.3.2.9 por_ccg_ha_secure_register_groups_override on page 298
16'h900	por_ccg_ha_unit_info	RO	4.3.2.10 por_ccg_ha_unit_info on page 299
16'h908	por_ccg_ha_unit_info2	RO	4.3.2.11 por_ccg_ha_unit_info2 on page 300
16'h910	por_ccg_ha_unit_info3	RO	4.3.2.12 por_ccg_ha_unit_info3 on page 301
16'h1F00	por_ccg_ha_agentid_to_linkid_reg0	RW	4.3.2.13 por_ccg_ha_agentid_to_linkid_reg0 on page 302
16'h1F08	por_ccg_ha_agentid_to_linkid_reg1	RW	4.3.2.14 por_ccg_ha_agentid_to_linkid_reg1 on page 303
16'h1F10	por_ccg_ha_agentid_to_linkid_reg2	RW	4.3.2.15 por_ccg_ha_agentid_to_linkid_reg2 on page 305
16'h1F18	por_ccg_ha_agentid_to_linkid_reg3	RW	4.3.2.16 por_ccg_ha_agentid_to_linkid_reg3 on page 306
16'h1F20	por_ccg_ha_agentid_to_linkid_reg4	RW	4.3.2.17 por_ccg_ha_agentid_to_linkid_reg4 on page 307
16'h1F28	por_ccg_ha_agentid_to_linkid_reg5	RW	4.3.2.18 por_ccg_ha_agentid_to_linkid_reg5 on page 309
16'h1F30	por_ccg_ha_agentid_to_linkid_reg6	RW	4.3.2.19 por_ccg_ha_agentid_to_linkid_reg6 on page 310
16'h1F38	por_ccg_ha_agentid_to_linkid_reg7	RW	4.3.2.20 por_ccg_ha_agentid_to_linkid_reg7 on page 311
16'h1FF8	por_ccg_ha_agentid_to_linkid_val	RW	4.3.2.21 por_ccg_ha_agentid_to_linkid_val on page 313
16'hC00 : 16'h13F8	por_ccg_ha_rnf_exp_raid_to_ldid_reg_0-255	RW	4.3.2.22 por_ccg_ha_rnf_exp_raid_to_ldid_reg_0-255 on page 314
16'h2000	por_ccg_ha_pmu_event_sel	RW	4.3.2.23 por_ccg_ha_pmu_event_sel on page 315
16'h1C00	por_ccg_ha_cxprtcl_link0_ctl	RW	4.3.2.24 por_ccg_ha_cxprtcl_link0_ctl on page 317
16'h1C08	por_ccg_ha_cxprtcl_link0_status	RO	4.3.2.25 por_ccg_ha_cxprtcl_link0_status on page 319
16'h1C10	por_ccg_ha_cxprtcl_link1_ctl	RW	4.3.2.26 por_ccg_ha_cxprtcl_link1_ctl on page 320
16'h1C18	por_ccg_ha_cxprtcl_link1_status	RO	4.3.2.27 por_ccg_ha_cxprtcl_link1_status on page 323
16'h1C20	por_ccg_ha_cxprtcl_link2_ctl	RW	4.3.2.28 por_ccg_ha_cxprtcl_link2_ctl on page 324
16'h1C28	por_ccg_ha_cxprtcl_link2_status	RO	4.3.2.29 por_ccg_ha_cxprtcl_link2_status on page 326

4.2.3 CCG_RA register summary

This section lists the CCG_RA registers used in CMN-700.

CCG_RA register summary

The following table shows the *CCG_RA* registers in offset order from the base memory address

Table 4-4: CCG_RA register summary

Offset	Name	Type	Description
16'h0	por_ccg_ra_node_info	RO	4.3.3.1 por_ccg_ra_node_info on page 328
16'h80	por_ccg_ra_child_info	RO	4.3.3.2 por_ccg_ra_child_info on page 329
16'h980	por_ccg_ra_secure_register_groups_override	RW	4.3.3.3 por_ccg_ra_secure_register_groups_override on page 330
16'h900	por_ccg_ra_unit_info	RO	4.3.3.4 por_ccg_ra_unit_info on page 331
16'hA00	por_ccg_ra_cfg_ctl	RW	4.3.3.5 por_ccg_ra_cfg_ctl on page 332
16'hA08	por_ccg_ra_aux_ctl	RW	4.3.3.6 por_ccg_ra_aux_ctl on page 334
16'hA18	por_ccg_ra_cbusy_limit_ctl	RW	4.3.3.7 por_ccg_ra_cbusy_limit_ctl on page 337

Offset	Name	Type	Description
16'hC00 : 16'hC38	por_ccg_ra_sam_addr_reg0-7on_reg0-7ndex	RW	4.3.3.8 por_ccg_ra_sam_addr_reg0-7on_reg0-7ndex on page 338
16'hD00	por_ccg_ra_agentid_to_linkid_val	RW	4.3.3.9 por_ccg_ra_agentid_to_linkid_val on page 339
16'hD10 : 16'hD48	por_ccg_ra_agent0-7d_to_l0-7nk0-7d_reg0-7ndex	RW	4.3.3.10 por_ccg_ra_agent0-7d_to_l0-7nk0-7d_reg0-7ndex on page 340
16'hE00 : 16'hEF8	por_ccg_ra_rn0-31_ld0-31d_to_exp_ra0-31d_reg0-31ndex	RW	4.3.3.11 por_ccg_ra_rn0-31_ld0-31d_to_exp_ra0-31d_reg0-31ndex on page 341
16'hF00 : 16'hFF8	por_ccg_ra_rnd_ld0-31d_to_exp_ra0-31d_reg0-31ndex	RW	4.3.3.12 por_ccg_ra_rnd_ld0-31d_to_exp_ra0-31d_reg0-31ndex on page 342
16'h1000 : 16'h13F8	por_ccg_ra_rnf_ld0-127d_to_exp_ra0-127d_reg0-127ndex	RW	4.3.3.13 por_ccg_ra_rnf_ld0-127d_to_exp_ra0-127d_reg0-127ndex on page 343
16'h1400 : 16'h1478	por_ccg_ra_ha_ld0-15d_to_exp_ra0-15d_reg0-15ndex	RW	4.3.3.14 por_ccg_ra_ha_ld0-15d_to_exp_ra0-15d_reg0-15ndex on page 345
16'h1480 : 16'h1480	por_ccg_ra_hns_ld0-0d_to_exp_ra0-0d_reg0-0ndex	RW	4.3.3.15 por_ccg_ra_hns_ld0-0d_to_exp_ra0-0d_reg0-0ndex on page 346
16'h1500 : 16'h18F8	por_ccg_ra_rnf_ld0-127d_to_ovrd_ld0-127d_reg0-127ndex	RW	4.3.3.16 por_ccg_ra_rnf_ld0-127d_to_ovrd_ld0-127d_reg0-127ndex on page 347
16'h2000	por_ccg_ra_pmu_event_sel	RW	4.3.3.17 por_ccg_ra_pmu_event_sel on page 348
16'h1C00	por_ccg_ra_ccprtcl_link0_ctl	RW	4.3.3.18 por_ccg_ra_ccprtcl_link0_ctl on page 350
16'h1C08	por_ccg_ra_ccprtcl_link0_status	RO	4.3.3.19 por_ccg_ra_ccprtcl_link0_status on page 353
16'h1C10	por_ccg_ra_ccprtcl_link1_ctl	RW	4.3.3.20 por_ccg_ra_ccprtcl_link1_ctl on page 354
16'h1C18	por_ccg_ra_ccprtcl_link1_status	RO	4.3.3.21 por_ccg_ra_ccprtcl_link1_status on page 357
16'h1C20	por_ccg_ra_ccprtcl_link2_ctl	RW	4.3.3.22 por_ccg_ra_ccprtcl_link2_ctl on page 358
16'h1C28	por_ccg_ra_ccprtcl_link2_status	RO	4.3.3.23 por_ccg_ra_ccprtcl_link2_status on page 361

4.2.4 CCLA register summary

This section lists the CCLA registers used in CMN-700.

CCLA register summary

The following table shows the *CCLA* registers in offset order from the base memory address

Table 4-5: CCLA register summary

Offset	Name	Type	Description
16'h0	por_ccla_node_info	RO	4.3.4.1 por_ccla_node_info on page 363
16'h80	por_ccla_child_info	RO	4.3.4.2 por_ccla_child_info on page 364
16'h988	por_ccla_secure_register_groups_override	RW	4.3.4.3 por_ccla_secure_register_groups_override on page 365
16'h910	por_ccla_unit_info	RO	4.3.4.4 por_ccla_unit_info on page 366

Offset	Name	Type	Description
16'hB00	por_ccla_cfg_ctl	RW	4.3.4.5 por_ccla_cfg_ctl on page 367
16'hB08	por_ccla_aux_ctl	RW	4.3.4.6 por_ccla_aux_ctl on page 369
16'hC00	por_ccla_ccix_prop_capabilities	RO	4.3.4.7 por_ccla_ccix_prop_capabilities on page 370
16'hC08	por_ccla_cxs_attr_capabilities	RO	4.3.4.8 por_ccla_cxs_attr_capabilities on page 372
16'hD00	por_ccla_permsg_pyld_0_63	RW	4.3.4.9 por_ccla_permsg_pyld_0_63 on page 374
16'hD08	por_ccla_permsg_pyld_64_127	RW	4.3.4.10 por_ccla_permsg_pyld_64_127 on page 374
16'hD10	por_ccla_permsg_pyld_128_191	RW	4.3.4.11 por_ccla_permsg_pyld_128_191 on page 375
16'hD18	por_ccla_permsg_pyld_192_255	RW	4.3.4.12 por_ccla_permsg_pyld_192_255 on page 376
16'hD20	por_ccla_permsg_ctl	RW	4.3.4.13 por_ccla_permsg_ctl on page 377
16'hD28	por_ccla_err_agent_id	RW	4.3.4.14 por_ccla_err_agent_id on page 378
16'hD30	por_ccla_agentid_to_portid_reg0	RW	4.3.4.15 por_ccla_agentid_to_portid_reg0 on page 379
16'hD38	por_ccla_agentid_to_portid_reg1	RW	4.3.4.16 por_ccla_agentid_to_portid_reg1 on page 380
16'hD40	por_ccla_agentid_to_portid_reg2	RW	4.3.4.17 por_ccla_agentid_to_portid_reg2 on page 382
16'hD48	por_ccla_agentid_to_portid_reg3	RW	4.3.4.18 por_ccla_agentid_to_portid_reg3 on page 383
16'hD50	por_ccla_agentid_to_portid_reg4	RW	4.3.4.19 por_ccla_agentid_to_portid_reg4 on page 384
16'hD58	por_ccla_agentid_to_portid_reg5	RW	4.3.4.20 por_ccla_agentid_to_portid_reg5 on page 386
16'hD60	por_ccla_agentid_to_portid_reg6	RW	4.3.4.21 por_ccla_agentid_to_portid_reg6 on page 387
16'hD68	por_ccla_agentid_to_portid_reg7	RW	4.3.4.22 por_ccla_agentid_to_portid_reg7 on page 388
16'hD70	por_ccla_agentid_to_portid_val	RW	4.3.4.23 por_ccla_agentid_to_portid_val on page 390
16'hD78	por_ccla_portfwd_en	RW	4.3.4.24 por_ccla_portfwd_en on page 391
16'hD80	por_ccla_portfwd_status	RO	4.3.4.25 por_ccla_portfwd_status on page 391
16'hD88	por_ccla_portfwd_req	RW	4.3.4.26 por_ccla_portfwd_req on page 392
16'hD90	por_ccla_linkid_to_hops	RW	4.3.4.27 por_ccla_linkid_to_hops on page 393
16'hE00	por_ccla_cxl_link_rx_credit_ctl	RW	4.3.4.28 por_ccla_cxl_link_rx_credit_ctl on page 394
16'hE08	por_ccla_cxl_link_rx_credit_return_stat	RO	4.3.4.29 por_ccla_cxl_link_rx_credit_return_stat on page 395
16'hE10	por_ccla_cxl_link_tx_credit_stat	RO	4.3.4.30 por_ccla_cxl_link_tx_credit_stat on page 396
16'hE50	por_ccla_cxl_security_policy	RW	4.3.4.31 por_ccla_cxl_security_policy on page 397
16'hE78	por_ccla_cxl_hdm_decoder_capability	RO	4.3.4.32 por_ccla_cxl_hdm_decoder_capability on page 398
16'hE80	por_ccla_cxl_hdm_decoder_global_control	RW	4.3.4.33 por_ccla_cxl_hdm_decoder_global_control on page 400
16'hE88	por_ccla_cxl_hdm_decoder_0_base_low	RWL	4.3.4.34 por_ccla_cxl_hdm_decoder_0_base_low on page 401
16'hE90	por_ccla_cxl_hdm_decoder_0_base_high	RWL	4.3.4.35 por_ccla_cxl_hdm_decoder_0_base_high on page 402
16'hE98	por_ccla_cxl_hdm_decoder_0_size_low	RWL	4.3.4.36 por_ccla_cxl_hdm_decoder_0_size_low on page 403
16'hEA0	por_ccla_cxl_hdm_decoder_0_size_high	RWL	4.3.4.37 por_ccla_cxl_hdm_decoder_0_size_high on page 404
16'hEA8	por_ccla_cxl_hdm_decoder_0_control	RWL	4.3.4.38 por_ccla_cxl_hdm_decoder_0_control on page 405
16'hEC0	por_ccla_cxl_hdm_decoder_0_dpa_skip_low	RWL	4.3.4.39 por_ccla_cxl_hdm_decoder_0_dpa_skip_low on page 406
16'hEC8	por_ccla_cxl_hdm_decoder_0_dpa_skip_high	RWL	4.3.4.40 por_ccla_cxl_hdm_decoder_0_dpa_skip_high on page 407
16'hED0	por_ccla_snoop_filter_group_id	RW	4.3.4.41 por_ccla_snoop_filter_group_id on page 408
16'hED8	por_ccla_snoop_filter_effective_size	RW	4.3.4.42 por_ccla_snoop_filter_effective_size on page 409
16'hEE0	por_ccla_dvsec_cxl_range_1_base_high	RWL	4.3.4.43 por_ccla_dvsec_cxl_range_1_base_high on page 410
16'hEE8	por_ccla_dvsec_cxl_range_1_base_low	RWL	4.3.4.44 por_ccla_dvsec_cxl_range_1_base_low on page 411

Offset	Name	Type	Description
16'hEF0	por_ccla_dvsec_cxl_range_2_base_high	RWL	4.3.4.45 por_ccla_dvsec_cxl_range_2_base_high on page 412
16'hEF8	por_ccla_dvsec_cxl_range_2_base_low	RWL	4.3.4.46 por_ccla_dvsec_cxl_range_2_base_low on page 413
16'hF00	por_ccla_dvsec_cxl_control	RWL	4.3.4.47 por_ccla_dvsec_cxl_control on page 414
16'hF08	por_ccla_dvsec_cxl_control2	RW	4.3.4.48 por_ccla_dvsec_cxl_control2 on page 415
16'hF10	por_ccla_dvsec_cxl_lock	RW	4.3.4.49 por_ccla_dvsec_cxl_lock on page 417
16'hF18	por_ccla_dvsec_flex_bus_port_control	RW	4.3.4.50 por_ccla_dvsec_flex_bus_port_control on page 418
16'hF40	por_ccla_err_capabilities_control	RW	4.3.4.51 por_ccla_err_capabilities_control on page 419
16'hF58	por_ccla_IDE_key_refresh_time_control	RW	4.3.4.52 por_ccla_IDE_key_refresh_time_control on page 420
16'hF60	por_ccla_IDE_truncation_transmit_delay_control	RW	4.3.4.53 por_ccla_IDE_truncation_transmit_delay_control on page 421
16'hF70	por_ccla_ll_to_ull_msg	RW	4.3.4.54 por_ccla_ll_to_ull_msg on page 422
16'hF80	por_ccla_cxl_timeout_isolation_control	RW	4.3.4.55 por_ccla_cxl_timeout_isolation_control on page 422
16'hF28	por_ccla_root_port_n_security_policy	RW	4.3.4.56 por_ccla_root_port_n_security_policy on page 424
16'hF30	por_ccla_root_port_n_id	RW	4.3.4.57 por_ccla_root_port_n_id on page 425
16'hE18	por_ccla_cxl_link_layer_defeature	RW	4.3.4.58 por_ccla_cxl_link_layer_defeature on page 426
16'hE20	por_ccla_ull_ctl	RW	4.3.4.59 por_ccla_ull_ctl on page 427
16'hE28	por_ccla_ull_status	RO	4.3.4.60 por_ccla_ull_status on page 428
16'hE30	por_ccla_cxl_ll_errinject_ctl	RW	4.3.4.61 por_ccla_cxl_ll_errinject_ctl on page 429
16'hE38	por_ccla_cxl_ll_errinject_stat	RO	4.3.4.62 por_ccla_cxl_ll_errinject_stat on page 430
16'hE40	por_ccla_cxl_viral_prop_en	RW	4.3.4.63 por_ccla_cxl_viral_prop_en on page 431
16'h2008	por_ccla_pmu_event_sel	RW	4.3.4.64 por_ccla_pmu_event_sel on page 432
16'h3000	por_ccla_errfr	RO	4.3.4.65 por_ccla_errfr on page 433
16'h3008	por_ccla_errctlr	RW	4.3.4.66 por_ccla_errctlr on page 435
16'h3010	por_ccla_errstatus	W1C	4.3.4.67 por_ccla_errstatus on page 436
16'h3018	por_ccla_erraddr	RW	4.3.4.68 por_ccla_erraddr on page 438
16'h3020	por_ccla_errmisc	RW	4.3.4.69 por_ccla_errmisc on page 439
16'h3100	por_ccla_errfr_NS	RO	4.3.4.70 por_ccla_errfr_NS on page 440
16'h3108	por_ccla_errctlr_NS	RW	4.3.4.71 por_ccla_errctlr_NS on page 441
16'h3110	por_ccla_errstatus_NS	W1C	4.3.4.72 por_ccla_errstatus_NS on page 442
16'h3118	por_ccla_erraddr_NS	RW	4.3.4.73 por_ccla_erraddr_NS on page 444
16'h3120	por_ccla_errmisc_NS	RW	4.3.4.74 por_ccla_errmisc_NS on page 445

4.2.5 Configuration manager register summary

This section lists the configuration manager registers used in CMN-700.

CFGM register summary

The following table shows the *CFGM* registers in offset order from the base memory address

Table 4-6: CFGM register summary

Offset	Name	Type	Description
16'h0	por_cfgm_node_info	RO	4.3.5.1 por_cfgm_node_info on page 446
16'h8	por_cfgm_periph_id_0_periph_id_1	RO	4.3.5.2 por_cfgm_periph_id_0_periph_id_1 on page 447
16'h10	por_cfgm_periph_id_2_periph_id_3	RO	4.3.5.3 por_cfgm_periph_id_2_periph_id_3 on page 448
16'h18	por_cfgm_periph_id_4_periph_id_5	RO	4.3.5.4 por_cfgm_periph_id_4_periph_id_5 on page 449
16'h20	por_cfgm_periph_id_6_periph_id_7	RO	4.3.5.5 por_cfgm_periph_id_6_periph_id_7 on page 450
16'h28	por_cfgm_component_id_0_component_id_1	RO	4.3.5.6 por_cfgm_component_id_0_component_id_1 on page 451
16'h30	por_cfgm_component_id_2_component_id_3	RO	4.3.5.7 por_cfgm_component_id_2_component_id_3 on page 452
16'h80	por_cfgm_child_info	RO	4.3.5.8 por_cfgm_child_info on page 453
16'h980	por_cfgm_secure_access	RW	4.3.5.9 por_cfgm_secure_access on page 454
16'h988	por_cfgm_secure_register_groups_override	RW	4.3.5.10 por_cfgm_secure_register_groups_override on page 455
16'h3000 : 16'h3038	por_cfgm_errgsr_mxp_0-7	RO	4.3.5.11 por_cfgm_errgsr_mxp_0-7 on page 456
16'h3040 : 16'h3078	por_cfgm_errgsr_mxp_0-7_NS	RO	4.3.5.12 por_cfgm_errgsr_mxp_0-7_NS on page 457
16'h3080 : 16'h30B8	por_cfgm_errgsr_hni_0-7	RO	4.3.5.13 por_cfgm_errgsr_hni_0-7 on page 458
16'h30C0 : 16'h30F8	por_cfgm_errgsr_hni_0-7_NS	RO	4.3.5.14 por_cfgm_errgsr_hni_0-7_NS on page 459
16'h3100 : 16'h3138	por_cfgm_errgsr_hnf_0-7	RO	4.3.5.15 por_cfgm_errgsr_hnf_0-7 on page 459
16'h3140 : 16'h3178	por_cfgm_errgsr_hnf_0-7_NS	RO	4.3.5.16 por_cfgm_errgsr_hnf_0-7_NS on page 460
16'h3180 : 16'h31B8	por_cfgm_errgsr_sbsx_0-7	RO	4.3.5.17 por_cfgm_errgsr_sbsx_0-7 on page 461
16'h31C0 : 16'h31F8	por_cfgm_errgsr_sbsx_0-7_NS	RO	4.3.5.18 por_cfgm_errgsr_sbsx_0-7_NS on page 462
16'h3200 : 16'h3238	por_cfgm_errgsr_cxg_0-7	RO	4.3.5.19 por_cfgm_errgsr_cxg_0-7 on page 463
16'h3240 : 16'h3278	por_cfgm_errgsr_cxg_0-7_NS	RO	4.3.5.20 por_cfgm_errgsr_cxg_0-7_NS on page 464
16'h3280 : 16'h32B8	por_cfgm_errgsr_mtsx_0-7	RO	4.3.5.21 por_cfgm_errgsr_mtsx_0-7 on page 465
16'h32C0 : 16'h32F8	por_cfgm_errgsr_mtsx_0-7_NS	RO	4.3.5.22 por_cfgm_errgsr_mtsx_0-7_NS on page 466
16'h3FA8	por_cfgm_errdevaff	RO	4.3.5.23 por_cfgm_errdevaff on page 467
16'h3FB8	por_cfgm_errdevarch	RO	4.3.5.24 por_cfgm_errdevarch on page 468
16'h3FC8	por_cfgm_erridr	RO	4.3.5.25 por_cfgm_erridr on page 469
16'h3FD0	por_cfgm_errpidr45	RO	4.3.5.26 por_cfgm_errpidr45 on page 470
16'h3FD8	por_cfgm_errpidr67	RO	4.3.5.27 por_cfgm_errpidr67 on page 471
16'h3FE0	por_cfgm_errpidr01	RO	4.3.5.28 por_cfgm_errpidr01 on page 472
16'h3FE8	por_cfgm_errpidr23	RO	4.3.5.29 por_cfgm_errpidr23 on page 473

Offset	Name	Type	Description
16'h3FF0	por_cfgm_errcidr01	RO	4.3.5.30 por_cfgm_errcidr01 on page 474
16'h3FF8	por_cfgm_errcidr23	RO	4.3.5.31 por_cfgm_errcidr23 on page 475
16'h900	por_info_global	RO	4.3.5.32 por_info_global on page 476
16'h908	por_info_global_1	RO	4.3.5.33 por_info_global_1 on page 478
16'h1C00	por_ppu_int_enable	RW	4.3.5.34 por_ppu_int_enable on page 479
16'h1C08	por_ppu_int_enable_1	RW	4.3.5.35 por_ppu_int_enable_1 on page 480
16'h1C10	por_ppu_int_status	W1C	4.3.5.36 por_ppu_int_status on page 481
16'h1C18	por_ppu_int_status_1	W1C	4.3.5.37 por_ppu_int_status_1 on page 482
16'h1C20	por_ppu_qactive_hyst	RW	4.3.5.38 por_ppu_qactive_hyst on page 483
16'h1C28	por_mpam_s_err_int_status	W1C	4.3.5.39 por_mpam_s_err_int_status on page 483
16'h1C30	por_mpam_s_err_int_status_1	W1C	4.3.5.40 por_mpam_s_err_int_status_1 on page 484
16'h1C38	por_mpam_ns_err_int_status	W1C	4.3.5.41 por_mpam_ns_err_int_status on page 485
16'h1C40	por_mpam_ns_err_int_status_1	W1C	4.3.5.42 por_mpam_ns_err_int_status_1 on page 486
16'h100 : 16'h8F8	por_cfgm_child_pointer_0-255	RO	4.3.5.43 por_cfgm_child_pointer_0-255 on page 487

4.2.6 CXLAPB register summary

This section lists the CXLAPB registers used in CMN-700

CXLAPB register summary

The following table shows the *CXLAPB* registers in offset order from the base memory address

Table 4-7: por_cxlapb_cfg register summary

Offset	Name	Type	Description
16'h1110	por_cxlapb_link_rx_credit_ctl	RW	4.3.6.1 por_cxlapb_link_rx_credit_ctl on page 488
16'h1118	por_cxlapb_link_rx_credit_return_stat	RO	4.3.6.2 por_cxlapb_link_rx_credit_return_stat on page 489
16'h1120	por_cxlapb_link_tx_credit_stat	RO	4.3.6.3 por_cxlapb_link_tx_credit_stat on page 490
16'h1060	por_cxlapb_cxl_security_policy	RW	4.3.6.4 por_cxlapb_cxl_security_policy on page 491
16'h1200	por_cxlapb_cxl_hdm_decoder_capability	RO	4.3.6.5 por_cxlapb_cxl_hdm_decoder_capability on page 492
16'h1204	por_cxlapb_cxl_hdm_decoder_global_control	RW	4.3.6.6 por_cxlapb_cxl_hdm_decoder_global_control on page 494
16'h1210	por_cxlapb_cxl_hdm_decoder_0_base_low	RWL	4.3.6.7 por_cxlapb_cxl_hdm_decoder_0_base_low on page 495
16'h1214	por_cxlapb_cxl_hdm_decoder_0_base_high	RWL	4.3.6.8 por_cxlapb_cxl_hdm_decoder_0_base_high on page 495
16'h1218	por_cxlapb_cxl_hdm_decoder_0_size_low	RWL	4.3.6.9 por_cxlapb_cxl_hdm_decoder_0_size_low on page 496
16'h121C	por_cxlapb_cxl_hdm_decoder_0_size_high	RWL	4.3.6.10 por_cxlapb_cxl_hdm_decoder_0_size_high on page 497
16'h1220	por_cxlapb_cxl_hdm_decoder_0_control	RWL	4.3.6.11 por_cxlapb_cxl_hdm_decoder_0_control on page 498
16'h1224	por_cxlapb_cxl_hdm_decoder_0_dpa_skip_low	RWL	4.3.6.12 por_cxlapb_cxl_hdm_decoder_0_dpa_skip_low on page 499
16'h1228	por_cxlapb_cxl_hdm_decoder_0_dpa_skip_high	RWL	4.3.6.13 por_cxlapb_cxl_hdm_decoder_0_dpa_skip_high on page 500
16'h1800	por_cxlapb_snoop_filter_group_id	RO	4.3.6.14 por_cxlapb_snoop_filter_group_id on page 501
16'h1804	por_cxlapb_snoop_filter_effective_size	RO	4.3.6.15 por_cxlapb_snoop_filter_effective_size on page 502

Offset	Name	Type	Description
16'h120	por_cxlapb_dvsec_cxl_range_1_base_high	RWL	4.3.6.16 por_cxlapb_dvsec_cxl_range_1_base_high on page 503
16'h124	por_cxlapb_dvsec_cxl_range_1_base_low	RWL	4.3.6.17 por_cxlapb_dvsec_cxl_range_1_base_low on page 504
16'h130	por_cxlapb_dvsec_cxl_range_2_base_high	RWL	4.3.6.18 por_cxlapb_dvsec_cxl_range_2_base_high on page 505
16'h134	por_cxlapb_dvsec_cxl_range_2_base_low	RWL	4.3.6.19 por_cxlapb_dvsec_cxl_range_2_base_low on page 506
16'h10C	por_cxlapb_dvsec_cxl_control	RWL	4.3.6.20 por_cxlapb_dvsec_cxl_control on page 507
16'h110	por_cxlapb_dvsec_cxl_control2	RW	4.3.6.21 por_cxlapb_dvsec_cxl_control2 on page 508
16'h114	por_cxlapb_dvsec_cxl_lock	RW	4.3.6.22 por_cxlapb_dvsec_cxl_lock on page 509
16'h20C	por_cxlapb_dvsec_flex_bus_port_control	RW	4.3.6.23 por_cxlapb_dvsec_flex_bus_port_control on page 510
16'h1014	por_cxlapb_err_capabilities_control	RW	4.3.6.24 por_cxlapb_err_capabilities_control on page 511
16'h18	por_cxlapb_IDE_key_refresh_time_control	RW	4.3.6.25 por_cxlapb_IDE_key_refresh_time_control on page 512
16'h24	por_cxlapb_IDE_truncation_transmit_delay_control	RW	4.3.6.26 por_cxlapb_IDE_truncation_transmit_delay_control on page 513
16'h0	por_cxlapb_ll_to_ull_msg	RW	4.3.6.27 por_cxlapb_ll_to_ull_msg on page 514
16'h8	por_cxlapb_cxl_timeout_isolation_control	RW	4.3.6.28 por_cxlapb_cxl_timeout_isolation_control on page 515
16'h1130	por_cxlapb_link_layer_defeature	RW	4.3.6.29 por_cxlapb_link_layer_defeature on page 516

4.2.7 Debug and trace register summary

This section lists the debug and trace registers used in CMN-700.

DT register summary

The following table shows the *DT* registers in offset order from the base memory address

Table 4-8: DT register summary

Offset	Name	Type	Description
16'h0	por_dt_node_info	RO	4.3.7.1 por_dt_node_info on page 517
16'h80	por_dt_child_info	RO	4.3.7.2 por_dt_child_info on page 518
16'h980	por_dt_secure_access	RW	4.3.7.3 por_dt_secure_access on page 519
16'hA00	por_dt_dtc_ctl	RW	4.3.7.4 por_dt_dtc_ctl on page 520
16'hA10	por_dt_trigger_status	RO	4.3.7.5 por_dt_trigger_status on page 522
16'hA20	por_dt_trigger_status_clr	WO	4.3.7.6 por_dt_trigger_status_clr on page 522
16'hA30	por_dt_trace_control	RW	4.3.7.7 por_dt_trace_control on page 523
16'hA48	por_dt_traceid	RW	4.3.7.8 por_dt_traceid on page 524
16'h2000	por_dt_pmevcntAB	RW	4.3.7.9 por_dt_pmevcntAB on page 525
16'h2010	por_dt_pmevcntCD	RW	4.3.7.10 por_dt_pmevcntCD on page 526
16'h2020	por_dt_pmevcntEF	RW	4.3.7.11 por_dt_pmevcntEF on page 527
16'h2030	por_dt_pmevcntGH	RW	4.3.7.12 por_dt_pmevcntGH on page 528
16'h2040	por_dt_pmccntr	RW	4.3.7.13 por_dt_pmccntr on page 529
16'h2050	por_dt_pmevcntsrAB	RW	4.3.7.14 por_dt_pmevcntsrAB on page 529
16'h2060	por_dt_pmevcntsrCD	RW	4.3.7.15 por_dt_pmevcntsrCD on page 530
16'h2070	por_dt_pmevcntsrEF	RW	4.3.7.16 por_dt_pmevcntsrEF on page 531

Offset	Name	Type	Description
16'h2080	por_dt_pmevcntsrGH	RW	4.3.7.17 por_dt_pmevcntsrGH on page 532
16'h2090	por_dt_pmccntrsr	RW	4.3.7.18 por_dt_pmccntrsr on page 533
16'h2100	por_dt_pmcr	RW	4.3.7.19 por_dt_pmcr on page 534
16'h2118	por_dt_pmovsr	RO	4.3.7.20 por_dt_pmovsr on page 535
16'h2120	por_dt_pmovsr_clr	WO	4.3.7.21 por_dt_pmovsr_clr on page 536
16'h2128	por_dt_pmssr	RO	4.3.7.22 por_dt_pmssr on page 537
16'h2130	por_dt_pmsrr	WO	4.3.7.23 por_dt_pmsrr on page 538
16'hFA0	por_dt_claim	RW	4.3.7.24 por_dt_claim on page 538
16'hFA8	por_dt_devaff	RO	4.3.7.25 por_dt_devaff on page 539
16'hFB0	por_dt_lsr	RO	4.3.7.26 por_dt_lsr on page 540
16'hFB8	por_dt_authstatus_devarch	RO	4.3.7.27 por_dt_authstatus_devarch on page 541
16'hFC0	por_dt_devid	RO	4.3.7.28 por_dt_devid on page 542
16'hFC8	por_dt_devtype	RO	4.3.7.29 por_dt_devtype on page 543
16'hFD0	por_dt_pidr45	RO	4.3.7.30 por_dt_pidr45 on page 544
16'hFD8	por_dt_pidr67	RO	4.3.7.31 por_dt_pidr67 on page 545
16'hFE0	por_dt_pidr01	RO	4.3.7.32 por_dt_pidr01 on page 546
16'hFE8	por_dt_pidr23	RO	4.3.7.33 por_dt_pidr23 on page 547
16'hFF0	por_dt_cidr01	RO	4.3.7.34 por_dt_cidr01 on page 548
16'hFF8	por_dt_cidr23	RO	4.3.7.35 por_dt_cidr23 on page 549

4.2.8 DN register summary

This section lists the DN registers used in CMN-700.

DN register summary

The following table shows the *DN* registers in offset order from the base memory address

Table 4-9: DN register summary

Offset	Name	Type	Description
16'h0	por_dn_node_info	RO	4.3.8.1 por_dn_node_info on page 550
16'h80	por_dn_child_info	RO	4.3.8.2 por_dn_child_info on page 551
16'h900	por_dn_build_info	RO	4.3.8.3 por_dn_build_info on page 552
16'h980	por_dn_secure_register_groups_override	RW	4.3.8.4 por_dn_secure_register_groups_override on page 553
16'hA00	por_dn_cfg_ctl	RW	4.3.8.5 por_dn_cfg_ctl on page 554
16'hA08	por_dn_aux_ctl	RW	4.3.8.6 por_dn_aux_ctl on page 555
16'hC00 + #{56*index}	por_dn_vmf0-15_ctrl	RW	4.3.8.7 por_dn_vmf0-15_ctrl on page 556
16'hC00 + #{56*index + 8}	por_dn_vmf0-15_rnf0	RW	4.3.8.8 por_dn_vmf0-15_rnf0 on page 557
16'hC00 + #{56*index + 16}	por_dn_vmf0-15_rnf1	RW	4.3.8.9 por_dn_vmf0-15_rnf1 on page 558

Offset	Name	Type	Description
16'hC00 + #{56*index + 24}	por_dn_vmf0-15_rnf2	RW	4.3.8.10 por_dn_vmf0-15_rnf2 on page 559
16'hC00 + #{56*index + 32}	por_dn_vmf0-15_rnf3	RW	4.3.8.11 por_dn_vmf0-15_rnf3 on page 560
16'hC00 + #{56*index + 40}	por_dn_vmf0-15_rnd0	RW	4.3.8.12 por_dn_vmf0-15_rnd0 on page 561
16'hC00 + #{56*index + 48}	por_dn_vmf0-15_cxra	RW	4.3.8.13 por_dn_vmf0-15_cxra on page 562
16'hF80 : 16'hF98	por_dn_domain_rnf0-3	RW	4.3.8.14 por_dn_domain_rnf0-3 on page 563
16'hFA0	por_dn_domain_rnd0	RW	4.3.8.15 por_dn_domain_rnd0 on page 564
16'hFA8	por_dn_domain_cxra	RW	4.3.8.16 por_dn_domain_cxra on page 565
16'hFB0 : 16'h1028	por_dn_vmf0-15_rnd1	RW	4.3.8.17 por_dn_vmf0-15_rnd1 on page 566
16'h1030	por_dn_domain_rnd1	RW	4.3.8.18 por_dn_domain_rnd1 on page 567
16'h2000	por_dn_pmu_event_sel	RW	4.3.8.19 por_dn_pmu_event_sel on page 568

4.2.9 HN-F register summary

This section lists the HN-F registers used in CMN-700.

HN-F register summary

The following table shows the *HN-F* registers in offset order from the base memory address

Table 4-10: HN-F register summary

Offset	Name	Type	Description
16'h0	cmn_hns_node_info	RO	4.3.10.1 cmn_hns_node_info on page 599
16'h80	cmn_hns_child_info	RO	4.3.10.2 cmn_hns_child_info on page 600
16'h980	cmn_hns_secure_register_groups_override	RW	4.3.10.3 cmn_hns_secure_register_groups_override on page 601
16'h900	cmn_hns_unit_info	RO	4.3.10.4 cmn_hns_unit_info on page 602
16'h908	cmn_hns_unit_info_1	RO	4.3.10.5 cmn_hns_unit_info_1 on page 605
16'hA00	cmn_hns_cfg_ctl	RW	4.3.10.6 cmn_hns_cfg_ctl on page 606
16'hA08	cmn_hns_aux_ctl	RW	4.3.10.7 cmn_hns_aux_ctl on page 609
16'hA10	cmn_hns_aux_ctl_1	RW	4.3.10.8 cmn_hns_aux_ctl_1 on page 613
16'hA18	cmn_hns_cbusy_limit_ctl	RW	4.3.10.9 cmn_hns_cbusy_limit_ctl on page 616
16'hA20	cmn_hns_txrsp_arb_weight_ctl	RW	4.3.10.10 cmn_hns_txrsp_arb_weight_ctl on page 617
16'hA28	cmn_hns_cbusy_mode_ctl	RW	4.3.10.11 cmn_hns_cbusy_mode_ctl on page 618
16'hA30	cmn_hns_lbt_cfg_ctl	RW	4.3.10.12 cmn_hns_lbt_cfg_ctl on page 620
16'hA38	cmn_hns_lbt_aux_ctl	RW	4.3.10.13 cmn_hns_lbt_aux_ctl on page 621
16'h1C00	cmn_hns_ppu_pwpr	RW	4.3.10.14 cmn_hns_ppu_pwpr on page 624
16'h1C08	cmn_hns_ppu_pwsr	RO	4.3.10.15 cmn_hns_ppu_pwsr on page 625
16'h1C14	cmn_hns_ppu_misr	RO	4.3.10.16 cmn_hns_ppu_misr on page 627
16'h2BB0	cmn_hns_ppu_idr0	RO	4.3.10.17 cmn_hns_ppu_idr0 on page 628

Offset	Name	Type	Description
16'h2BB4	cmn_hns_ppu_idr1	RO	4.3.10.18 cmn_hns_ppu_idr1 on page 630
16'h2BC8	cmn_hns_ppu_iidr	RO	4.3.10.19 cmn_hns_ppu_iidr on page 630
16'h2BCC	cmn_hns_ppu_aidr	RO	4.3.10.20 cmn_hns_ppu_aidr on page 631
16'h1D00	cmn_hns_ppu_dyn_ret_threshold	RW	4.3.10.21 cmn_hns_ppu_dyn_ret_threshold on page 632
16'hA80	cmn_hns_qos_band	RO	4.3.10.22 cmn_hns_qos_band on page 633
16'h3000	cmn_hns_errfr	RO	4.3.10.23 cmn_hns_errfr on page 635
16'h3008	cmn_hns_errctlr	RW	4.3.10.24 cmn_hns_errctlr on page 636
16'h3010	cmn_hns_errstatus	W1C	4.3.10.25 cmn_hns_errstatus on page 637
16'h3018	cmn_hns_erraddr	RW	4.3.10.26 cmn_hns_erraddr on page 639
16'h3020	cmn_hns_errmisc	RW	4.3.10.27 cmn_hns_errmisc on page 640
16'h3030	cmn_hns_err_inj	RW	4.3.10.28 cmn_hns_err_inj on page 642
16'h3038	cmn_hns_byte_par_err_inj	WO	4.3.10.29 cmn_hns_byte_par_err_inj on page 643
16'h3100	cmn_hns_errfr_NS	RO	4.3.10.30 cmn_hns_errfr_NS on page 644
16'h3108	cmn_hns_errctlr_NS	RW	4.3.10.31 cmn_hns_errctlr_NS on page 645
16'h3110	cmn_hns_errstatus_NS	W1C	4.3.10.32 cmn_hns_errstatus_NS on page 646
16'h3118	cmn_hns_erraddr_NS	RW	4.3.10.33 cmn_hns_erraddr_NS on page 648
16'h3120	cmn_hns_errmisc_NS	RW	4.3.10.34 cmn_hns_errmisc_NS on page 649
16'hC00	cmn_hns_slc_lock_ways	RW	4.3.10.35 cmn_hns_slc_lock_ways on page 650
16'hC08	cmn_hns_slc_lock_base0	RW	4.3.10.36 cmn_hns_slc_lock_base0 on page 651
16'hC10	cmn_hns_slc_lock_base1	RW	4.3.10.37 cmn_hns_slc_lock_base1 on page 652
16'hC18	cmn_hns_slc_lock_base2	RW	4.3.10.38 cmn_hns_slc_lock_base2 on page 653
16'hC20	cmn_hns_slc_lock_base3	RW	4.3.10.39 cmn_hns_slc_lock_base3 on page 654
16'hC28	cmn_hns_rni_region_vec	RW	4.3.10.40 cmn_hns_rni_region_vec on page 655
16'hC30	cmn_hns_rnd_region_vec	RW	4.3.10.41 cmn_hns_rnd_region_vec on page 656
16'hC38	cmn_hns_rnf_region_vec	RW	4.3.10.42 cmn_hns_rnf_region_vec on page 657
16'hC40	cmn_hns_rnf_region_vec1	RW	4.3.10.43 cmn_hns_rnf_region_vec1 on page 658
16'hC48	cmn_hns_slcway_partition0_rnf_vec	RW	4.3.10.44 cmn_hns_slcway_partition0_rnf_vec on page 659
16'hC50	cmn_hns_slcway_partition1_rnf_vec	RW	4.3.10.45 cmn_hns_slcway_partition1_rnf_vec on page 660
16'hC58	cmn_hns_slcway_partition2_rnf_vec	RW	4.3.10.46 cmn_hns_slcway_partition2_rnf_vec on page 661
16'hC60	cmn_hns_slcway_partition3_rnf_vec	RW	4.3.10.47 cmn_hns_slcway_partition3_rnf_vec on page 662
16'hCB0	cmn_hns_slcway_partition0_rnf_vec1	RW	4.3.10.48 cmn_hns_slcway_partition0_rnf_vec1 on page 663
16'hCB8	cmn_hns_slcway_partition1_rnf_vec1	RW	4.3.10.49 cmn_hns_slcway_partition1_rnf_vec1 on page 664
16'hCC0	cmn_hns_slcway_partition2_rnf_vec1	RW	4.3.10.50 cmn_hns_slcway_partition2_rnf_vec1 on page 665
16'hCC8	cmn_hns_slcway_partition3_rnf_vec1	RW	4.3.10.51 cmn_hns_slcway_partition3_rnf_vec1 on page 666

Offset	Name	Type	Description
16'hC68	cmn_hns_slcway_partition0_rni_vec	RW	4.3.10.52 cmn_hns_slcway_partition0_rni_vec on page 667
16'hC70	cmn_hns_slcway_partition1_rni_vec	RW	4.3.10.53 cmn_hns_slcway_partition1_rni_vec on page 668
16'hC78	cmn_hns_slcway_partition2_rni_vec	RW	4.3.10.54 cmn_hns_slcway_partition2_rni_vec on page 669
16'hC80	cmn_hns_slcway_partition3_rni_vec	RW	4.3.10.55 cmn_hns_slcway_partition3_rni_vec on page 670
16'hC88	cmn_hns_slcway_partition0_rnd_vec	RW	4.3.10.56 cmn_hns_slcway_partition0_rnd_vec on page 671
16'hC90	cmn_hns_slcway_partition1_rnd_vec	RW	4.3.10.57 cmn_hns_slcway_partition1_rnd_vec on page 672
16'hC98	cmn_hns_slcway_partition2_rnd_vec	RW	4.3.10.58 cmn_hns_slcway_partition2_rnd_vec on page 673
16'hCA0	cmn_hns_slcway_partition3_rnd_vec	RW	4.3.10.59 cmn_hns_slcway_partition3_rnd_vec on page 674
16'hCA8	cmn_hns_rn_region_lock	RW	4.3.10.60 cmn_hns_rn_region_lock on page 675
16'hCD0	cmn_hns_sf_cxg_blocked_ways	RW	4.3.10.61 cmn_hns_sf_cxg_blocked_ways on page 676
16'hCE0	cmn_hns_cxg_ha_metadata_exclusion_list	RW	4.3.10.62 cmn_hns_cxg_ha_metadata_exclusion_list on page 677
16'hCD8	cmn_hns_cxg_ha_smp_exclusion_list	RW	4.3.10.63 cmn_hns_cxg_ha_smp_exclusion_list on page 678
16'hCF0	hn_sam_hash_addr_mask_reg	RW	4.3.10.64 hn_sam_hash_addr_mask_reg on page 679
16'hCF8	hn_sam_region_cmp_addr_mask_reg	RW	4.3.10.65 hn_sam_region_cmp_addr_mask_reg on page 680
16'hD48	cmn_hns_sam_cfg1_def_hashed_region	RW	4.3.10.66 cmn_hns_sam_cfg1_def_hashed_region on page 681
16'hD50	cmn_hns_sam_cfg2_def_hashed_region	RW	4.3.10.67 cmn_hns_sam_cfg2_def_hashed_region on page 682
16'hD00	cmn_hns_sam_control	RW	4.3.10.68 cmn_hns_sam_control on page 683
16'hD28	cmn_hns_sam_control2	RW	4.3.10.69 cmn_hns_sam_control2 on page 685
16'hD08	cmn_hns_sam_memregion0	RW	4.3.10.70 cmn_hns_sam_memregion0 on page 686
16'hD38	cmn_hns_sam_memregion0_end_addr	RW	4.3.10.71 cmn_hns_sam_memregion0_end_addr on page 687
16'hD10	cmn_hns_sam_memregion1	RW	4.3.10.72 cmn_hns_sam_memregion1 on page 688
16'hD40	cmn_hns_sam_memregion1_end_addr	RW	4.3.10.73 cmn_hns_sam_memregion1_end_addr on page 689
16'hD18	cmn_hns_sam_sn_properties	RW	4.3.10.74 cmn_hns_sam_sn_properties on page 690
16'hD20	cmn_hns_sam_6sn_nodeid	RW	4.3.10.75 cmn_hns_sam_6sn_nodeid on page 693
16'hCE8	cmn_hns_sam_sn_properties1	RW	4.3.10.76 cmn_hns_sam_sn_properties1 on page 695
16'hD30	cmn_hns_sam_sn_properties2	RW	4.3.10.77 cmn_hns_sam_sn_properties2 on page 697
16'hF80 : 16'hFA0	cmn_hns_cml_port_aggr_grp0-4_add_mask	RW	4.3.10.78 cmn_hns_cml_port_aggr_grp0-4_add_mask on page 699
16'h6028 : 16'h60F8	cmn_hns_cml_port_aggr_grp5-31_add_mask	RW	4.3.10.79 cmn_hns_cml_port_aggr_grp5-31_add_mask on page 700

Offset	Name	Type	Description
{0-1} 16'hFB0 : 16'hFB8	cmn_hns_cml_port_aggr_grp_reg0-12	RW	4.3.10.80 cmn_hns_cml_port_aggr_grp_reg0-12 on page 701
{2-12} 16'h6110 : 16'h6160			
16'hFD0	cmn_hns_cml_port_aggr_ctrl_reg	RW	4.3.10.81 cmn_hns_cml_port_aggr_ctrl_reg on page 702
16'h6208 : 16'h6230	cmn_hns_cml_port_aggr_ctrl_reg1-6	RW	4.3.10.82 cmn_hns_cml_port_aggr_ctrl_reg1-6 on page 704
16'hF50	cmn_hns_abf_lo_addr	RW	4.3.10.83 cmn_hns_abf_lo_addr on page 707
16'hF58	cmn_hns_abf_hi_addr	RW	4.3.10.84 cmn_hns_abf_hi_addr on page 708
16'hF60	cmn_hns_abf_pr	RW	4.3.10.85 cmn_hns_abf_pr on page 709
16'hF68	cmn_hns_abf_sr	RO	4.3.10.86 cmn_hns_abf_sr on page 710
16'h1000	cmn_hns_cbusy_write_limit_ctl	RW	4.3.10.87 cmn_hns_cbusy_write_limit_ctl on page 711
16'h1008	cmn_hns_cbusy_resp_ctl	RW	4.3.10.88 cmn_hns_cbusy_resp_ctl on page 712
16'h1010	cmn_hns_cbusy_sn_ctl	RW	4.3.10.89 cmn_hns_cbusy_sn_ctl on page 714
16'h1018	cmn_hns_lbt_cbusy_ctl	RW	4.3.10.90 cmn_hns_lbt_cbusy_ctl on page 715
16'h1020	cmn_hns_pocq_alloc_class_dedicated	RW	4.3.10.91 cmn_hns_pocq_alloc_class_dedicated on page 716
16'h1028	cmn_hns_pocq_alloc_class_max_allowed	RW	4.3.10.92 cmn_hns_pocq_alloc_class_max_allowed on page 718
16'h1030	cmn_hns_pocq_alloc_class_contended_min	RW	4.3.10.93 cmn_hns_pocq_alloc_class_contended_min on page 719
16'h1038	cmn_hns_pocq_alloc_misc_max_allowed	RW	4.3.10.94 cmn_hns_pocq_alloc_misc_max_allowed on page 720
16'h1040	cmn_hns_class_ctl	RW	4.3.10.95 cmn_hns_class_ctl on page 721
16'h1048	cmn_hns_pocq_qos_class_ctl	RW	4.3.10.96 cmn_hns_pocq_qos_class_ctl on page 722
16'h1050	cmn_hns_class_pocq_arb_weight_ctl	RW	4.3.10.97 cmn_hns_class_pocq_arb_weight_ctl on page 723
16'h1058	cmn_hns_class_retry_weight_ctl	RW	4.3.10.98 cmn_hns_class_retry_weight_ctl on page 724
16'h1060	cmn_hns_pocq_misc_retry_weight_ctl	RW	4.3.10.99 cmn_hns_pocq_misc_retry_weight_ctl on page 725
16'hFE0	cmn_hns_partner_scratch_reg0	RW	4.3.10.100 cmn_hns_partner_scratch_reg0 on page 727
16'hFE8	cmn_hns_partner_scratch_reg1	RW	4.3.10.101 cmn_hns_partner_scratch_reg1 on page 728
16'hB80	cmn_hns_cfg_slcsf_dbgrd	WO	4.3.10.102 cmn_hns_cfg_slcsf_dbgrd on page 729
16'hB88	cmn_hns_slc_cache_access_slc_tag	RO	4.3.10.103 cmn_hns_slc_cache_access_slc_tag on page 730
16'hB90	cmn_hns_slc_cache_access_slc_tag1	RO	4.3.10.104 cmn_hns_slc_cache_access_slc_tag1 on page 731
16'hB98	cmn_hns_slc_cache_access_slc_data	RO	4.3.10.105 cmn_hns_slc_cache_access_slc_data on page 732
16'hBC0	cmn_hns_slc_cache_access_slc_mte_tag	RO	4.3.10.106 cmn_hns_slc_cache_access_slc_mte_tag on page 733
16'hBA0	cmn_hns_slc_cache_access_sf_tag	RO	4.3.10.107 cmn_hns_slc_cache_access_sf_tag on page 734
16'hBA8	cmn_hns_slc_cache_access_sf_tag1	RO	4.3.10.108 cmn_hns_slc_cache_access_sf_tag1 on page 735

Offset	Name	Type	Description
16'hBB0	cmn_hns_slc_cache_access_sf_tag2	RO	4.3.10.109 cmn_hns_slc_cache_access_sf_tag2 on page 736
16'h2000	cmn_hns_pmu_event_sel	RW	4.3.10.110 cmn_hns_pmu_event_sel on page 737
16'h2008	cmn_hns_pmu_mpam_sel	RW	4.3.10.111 cmn_hns_pmu_mpam_sel on page 741
16'h2010 : 16'h2048	cmn_hns_pmu_mpam_pardid_mask0-7	RW	4.3.10.112 cmn_hns_pmu_mpam_pardid_mask0-7 on page 742
16'h3C00 + #{index}*32	cmn_hns_rn_cluster0-63_physid_reg0	RW	4.3.10.113 cmn_hns_rn_cluster0-63_physid_reg0 on page 743
16'h3C00 + #{index}*32	cmn_hns_rn_cluster64-127_physid_reg0	RW	4.3.10.114 cmn_hns_rn_cluster64-127_physid_reg0 on page 746
16'h3C08 + #{index}*32	cmn_hns_rn_cluster0-127_physid_reg1	RW	4.3.10.115 cmn_hns_rn_cluster0-127_physid_reg1 on page 748
16'h3C10 + #{index}*32	cmn_hns_rn_cluster0-127_physid_reg2	RW	4.3.10.116 cmn_hns_rn_cluster0-127_physid_reg2 on page 750
16'h3C18 + #{index}*32	cmn_hns_rn_cluster0-127_physid_reg3	RW	4.3.10.117 cmn_hns_rn_cluster0-127_physid_reg3 on page 752
16'h5010 : 16'h51F8	cmn_hns_sam_nonhash_cfg1_memregion2-63	RW	4.3.10.118 cmn_hns_sam_nonhash_cfg1_memregion2-63 on page 754
16'h5210 : 16'h53F8	cmn_hns_sam_nonhash_cfg2_memregion2-63	RW	4.3.10.119 cmn_hns_sam_nonhash_cfg2_memregion2-63 on page 755
16'h5400 : 16'h5478	cmn_hns_sam_htg_cfg1_memregion0-15	RW	4.3.10.120 cmn_hns_sam_htg_cfg1_memregion0-15 on page 756
16'h5480 : 16'h54F8	cmn_hns_sam_htg_cfg2_memregion0-15	RW	4.3.10.121 cmn_hns_sam_htg_cfg2_memregion0-15 on page 758
16'h5500 : 16'h5578	cmn_hns_sam_htg_cfg3_memregion0-15	RW	4.3.10.122 cmn_hns_sam_htg_cfg3_memregion0-15 on page 759
16'h5600 : 16'h5678	cmn_hns_sam_htg_sn_nodeid_reg0-15	RW	4.3.10.123 cmn_hns_sam_htg_sn_nodeid_reg0-15 on page 761
16'h5680 : 16'h56F8	cmn_hns_sam_htg_sn_attr0-15	RW	4.3.10.124 cmn_hns_sam_htg_sn_attr0-15 on page 762
16'h5700 : 16'h5718	cmn_hns_sam_ccg_sa_nodeid_reg0-3	RW	4.3.10.125 cmn_hns_sam_ccg_sa_nodeid_reg0-3 on page 766
16'h5740 : 16'h5758	cmn_hns_sam_ccg_sa_attr0-3	RW	4.3.10.126 cmn_hns_sam_ccg_sa_attr0-3 on page 767
16'h5780 : 16'h57B8	hns_generic_regs0-7	RW	4.3.10.127 hns_generic_regs0-7 on page 771
16'h5900	cmn_hns_pa2setaddr_slc	RW	4.3.10.128 cmn_hns_pa2setaddr_slc on page 772
16'h5908	cmn_hns_pa2setaddr_sf	RW	4.3.10.129 cmn_hns_pa2setaddr_sf on page 776
16'h5910	cmn_hns_pa2setaddr_flex_slc	RW	4.3.10.130 cmn_hns_pa2setaddr_flex_slc on page 780
16'h5918	cmn_hns_pa2setaddr_flex_sf	RW	4.3.10.131 cmn_hns_pa2setaddr_flex_sf on page 781
16'h7000 : 16'h70F8	lcn_hashed_tgt_grp_cfg1_region0-31	RW	4.3.10.132 lcn_hashed_tgt_grp_cfg1_region0-31 on page 782
16'h7100 : 16'h71F8	lcn_hashed_tgt_grp_cfg2_region0-31	RW	4.3.10.133 lcn_hashed_tgt_grp_cfg2_region0-31 on page 784
16'h7200 : 16'h72F8	lcn_hashed_target_grp_secondary_cfg1_reg0-31	RW	4.3.10.134 lcn_hashed_target_grp_secondary_cfg1_reg0-31 on page 785

Offset	Name	Type	Description
16'h7300 : 16'h73F8	lcn_hashed_target_grp_secondary_cfg2_reg0-31	RW	4.3.10.135 lcn_hashed_target_grp_secondary_cfg2_reg0-31 on page 787
16'h7400 : 16'h74F8	lcn_hashed_target_grp_hash_cntl_reg0-31	RW	4.3.10.136 lcn_hashed_target_grp_hash_cntl_reg0-31 on page 788
16'h7500 : 16'h7518	lcn_hashed_target_group_hn_count_reg0-3	RW	4.3.10.137 lcn_hashed_target_group_hn_count_reg0-3 on page 790
16'h7520 : 16'h7558	lcn_hashed_target_grp_cal_mode_reg0-7	RW	4.3.10.138 lcn_hashed_target_grp_cal_mode_reg0-7 on page 791
16'h7560 : 16'h7568	lcn_hashed_target_grp_hnf_cpa_en_reg0-1	RW	4.3.10.139 lcn_hashed_target_grp_hnf_cpa_en_reg0-1 on page 793
16'h7580 : 16'h75F8	lcn_hashed_target_grp_cpag_perhnf_reg0-15	RW	4.3.10.140 lcn_hashed_target_grp_cpag_perhnf_reg0-15 on page 794
16'h7700 : 16'h77F8	lcn_hashed_target_grp_compact_cpag_ctrl0-31	RW	4.3.10.141 lcn_hashed_target_grp_compact_cpag_ctrl0-31 on page 796
16'h7800 : 16'h78F8	lcn_hashed_target_grp_compact_hash_ctrl0-31	RW	4.3.10.142 lcn_hashed_target_grp_compact_hash_ctrl0-31 on page 797

4.2.10 HN-F MPAM_NS register summary

This section lists the HN-F MPAM_NS registers used in CMN-700.

HNF_MPAM_NS register summary

The following table shows the *HNF_MPAM_NS* registers in offset order from the base memory address

Table 4-11: HNF_MPAM_NS register summary

Offset	Name	Type	Description
16'h0	cmn_hns_mpam_ns_node_info	RO	4.3.11.1 cmn_hns_mpam_ns_node_info on page 799
16'h80	cmn_hns_mpam_ns_child_info	RO	4.3.11.2 cmn_hns_mpam_ns_child_info on page 800
16'h1000	cmn_hns_mpam_idr	RO	4.3.11.3 cmn_hns_mpam_idr on page 801
16'h1018	cmn_hns_mpam_iidr	RO	4.3.11.4 cmn_hns_mpam_iidr on page 802
16'h1020	cmn_hns_mpam_aidr	RO	4.3.11.5 cmn_hns_mpam_aidr on page 803
16'h1028	cmn_hns_mpam_impl_idr	RO	4.3.11.6 cmn_hns_mpam_impl_idr on page 804
16'h1030	cmn_hns_mpam_cpor_idr	RO	4.3.11.7 cmn_hns_mpam_cpor_idr on page 805
16'h1038	cmn_hns_mpam_ccap_idr	RO	4.3.11.8 cmn_hns_mpam_ccap_idr on page 806
16'h1040	cmn_hns_mpam_mbw_idr	RO	4.3.11.9 cmn_hns_mpam_mbw_idr on page 807
16'h1048	cmn_hns_mpam_pri_idr	RO	4.3.11.10 cmn_hns_mpam_pri_idr on page 808
16'h1050	cmn_hns_mpam_partid_nrw_idr	RO	4.3.11.11 cmn_hns_mpam_partid_nrw_idr on page 810
16'h1080	cmn_hns_mpam_msmon_idr	RO	4.3.11.12 cmn_hns_mpam_msmon_idr on page 811
16'h1088	cmn_hns_mpam_csumon_idr	RO	4.3.11.13 cmn_hns_mpam_csumon_idr on page 812
16'h1090	cmn_hns_mpam_mbwumon_idr	RO	4.3.11.14 cmn_hns_mpam_mbwumon_idr on page 813
16'h10F0	cmn_hns_ns_mpam_ecr	RW	4.3.11.15 cmn_hns_ns_mpam_ecr on page 814

Offset	Name	Type	Description
16'h10F8	cmn_hns_ns_mpam_esr	RW	4.3.11.16 cmn_hns_ns_mpam_esr on page 815
16'h1100	cmn_hns_ns_mpamcfg_part_sel	RW	4.3.11.17 cmn_hns_ns_mpamcfg_part_sel on page 816
16'h1108	cmn_hns_ns_mpamcfg_cmax	RW	4.3.11.18 cmn_hns_ns_mpamcfg_cmax on page 817
16'h1200	cmn_hns_ns_mpamcfg_mbw_min	RW	4.3.11.19 cmn_hns_ns_mpamcfg_mbw_min on page 818
16'h1208	cmn_hns_ns_mpamcfg_mbw_max	RW	4.3.11.20 cmn_hns_ns_mpamcfg_mbw_max on page 819
16'h1220	cmn_hns_ns_mpamcfg_mbw_winwd	RW	4.3.11.21 cmn_hns_ns_mpamcfg_mbw_winwd on page 820
16'h1400	cmn_hns_ns_mpamcfg_pri	RW	4.3.11.22 cmn_hns_ns_mpamcfg_pri on page 821
16'h1500	cmn_hns_ns_mpamcfg_mbw_prop	RW	4.3.11.23 cmn_hns_ns_mpamcfg_mbw_prop on page 823
16'h1600	cmn_hns_ns_mpamcfg_intpartid	RW	4.3.11.24 cmn_hns_ns_mpamcfg_intpartid on page 824
16'h1800	cmn_hns_ns_msmon_cfg_mon_sel	RW	4.3.11.25 cmn_hns_ns_msmon_cfg_mon_sel on page 825
16'h1808	cmn_hns_ns_msmon_capt_evnt	RW	4.3.11.26 cmn_hns_ns_msmon_capt_evnt on page 826
16'h1810	cmn_hns_ns_msmon_cfg_csu_flt	RW	4.3.11.27 cmn_hns_ns_msmon_cfg_csu_flt on page 827
16'h1818	cmn_hns_ns_msmon_cfg_csu_ctl	RW	4.3.11.28 cmn_hns_ns_msmon_cfg_csu_ctl on page 828
16'h1820	cmn_hns_ns_msmon_cfg_mbwu_flt	RW	4.3.11.29 cmn_hns_ns_msmon_cfg_mbwu_flt on page 830
16'h1828	cmn_hns_ns_msmon_cfg_mbwu_ctl	RW	4.3.11.30 cmn_hns_ns_msmon_cfg_mbwu_ctl on page 831
16'h1840	cmn_hns_ns_msmon_csu	RW	4.3.11.31 cmn_hns_ns_msmon_csu on page 833
16'h1848	cmn_hns_ns_msmon_csu_capture	RW	4.3.11.32 cmn_hns_ns_msmon_csu_capture on page 834
16'h1860	cmn_hns_ns_msmon_mbwu	RW	4.3.11.33 cmn_hns_ns_msmon_mbwu on page 835
16'h1868	cmn_hns_ns_msmon_mbwu_capture	RW	4.3.11.34 cmn_hns_ns_msmon_mbwu_capture on page 836
16'h2000	cmn_hns_ns_mpamcfg_cpbm	RW	4.3.11.35 cmn_hns_ns_mpamcfg_cpbm on page 837

4.2.11 HN-F MPAM_S register summary

This section lists the HN-F MPAM_S registers used in CMN-700.

HNF_MPAM_S register summary

The following table shows the *HNF_MPAM_S* registers in offset order from the base memory address

Table 4-12: HNF_MPAM_S register summary

Offset	Name	Type	Description
16'h0	cmn_hns_mpam_s_node_info	RO	4.3.12.1 cmn_hns_mpam_s_node_info on page 838
16'h80	cmn_hns_mpam_s_child_info	RO	4.3.12.2 cmn_hns_mpam_s_child_info on page 839
16'h980	cmn_hns_mpam_s_secure_register_groups_override	RW	4.3.12.3 cmn_hns_mpam_s_secure_register_groups_override on page 840
16'h1000	cmn_hns_s_mpam_idr	RO	4.3.12.4 cmn_hns_s_mpam_idr on page 841
16'h1008	cmn_hns_mpam_sidr	RO	4.3.12.5 cmn_hns_mpam_sidr on page 842
16'h1018	cmn_hns_s_mpam_iidr	RO	4.3.12.6 cmn_hns_s_mpam_iidr on page 843
16'h1020	cmn_hns_s_mpam_aidr	RO	4.3.12.7 cmn_hns_s_mpam_aidr on page 844
16'h1028	cmn_hns_s_mpam_impl_idr	RO	4.3.12.8 cmn_hns_s_mpam_impl_idr on page 845
16'h1030	cmn_hns_s_mpam_cpor_idr	RO	4.3.12.9 cmn_hns_s_mpam_cpor_idr on page 846

Offset	Name	Type	Description
16'h1038	cmn_hns_s_mpam_ccap_idr	RO	4.3.12.10 cmn_hns_s_mpam_ccap_idr on page 847
16'h1040	cmn_hns_s_mpam_mbw_idr	RO	4.3.12.11 cmn_hns_s_mpam_mbw_idr on page 848
16'h1048	cmn_hns_s_mpam_pri_idr	RO	4.3.12.12 cmn_hns_s_mpam_pri_idr on page 849
16'h1050	cmn_hns_s_mpam_partid_nrw_idr	RO	4.3.12.13 cmn_hns_s_mpam_partid_nrw_idr on page 851
16'h1080	cmn_hns_s_mpam_msmon_idr	RO	4.3.12.14 cmn_hns_s_mpam_msmon_idr on page 852
16'h1088	cmn_hns_s_mpam_csumon_idr	RO	4.3.12.15 cmn_hns_s_mpam_csumon_idr on page 853
16'h1090	cmn_hns_s_mpam_mbwumon_idr	RO	4.3.12.16 cmn_hns_s_mpam_mbwumon_idr on page 854
16'h10F0	cmn_hns_s_mpam_ecr	RW	4.3.12.17 cmn_hns_s_mpam_ecr on page 855
16'h10F8	cmn_hns_s_mpam_esr	RW	4.3.12.18 cmn_hns_s_mpam_esr on page 856
16'h1100	cmn_hns_s_mpamcfg_part_sel	RW	4.3.12.19 cmn_hns_s_mpamcfg_part_sel on page 857
16'h1108	cmn_hns_s_mpamcfg_cmax	RW	4.3.12.20 cmn_hns_s_mpamcfg_cmax on page 858
16'h1200	cmn_hns_s_mpamcfg_mbw_min	RW	4.3.12.21 cmn_hns_s_mpamcfg_mbw_min on page 859
16'h1208	cmn_hns_s_mpamcfg_mbw_max	RW	4.3.12.22 cmn_hns_s_mpamcfg_mbw_max on page 861
16'h1220	cmn_hns_s_mpamcfg_mbw_winwd	RW	4.3.12.23 cmn_hns_s_mpamcfg_mbw_winwd on page 862
16'h1400	cmn_hns_s_mpamcfg_pri	RW	4.3.12.24 cmn_hns_s_mpamcfg_pri on page 863
16'h1500	cmn_hns_s_mpamcfg_mbw_prop	RW	4.3.12.25 cmn_hns_s_mpamcfg_mbw_prop on page 864
16'h1600	cmn_hns_s_mpamcfg_intpartid	RW	4.3.12.26 cmn_hns_s_mpamcfg_intpartid on page 865
16'h1800	cmn_hns_s_msmon_cfg_mon_sel	RW	4.3.12.27 cmn_hns_s_msmon_cfg_mon_sel on page 866
16'h1808	cmn_hns_s_msmon_capt_evnt	RW	4.3.12.28 cmn_hns_s_msmon_capt_evnt on page 867
16'h1810	cmn_hns_s_msmon_cfg_csu_flt	RW	4.3.12.29 cmn_hns_s_msmon_cfg_csu_flt on page 868
16'h1818	cmn_hns_s_msmon_cfg_csu_ctl	RW	4.3.12.30 cmn_hns_s_msmon_cfg_csu_ctl on page 870
16'h1820	cmn_hns_s_msmon_cfg_mbwu_flt	RW	4.3.12.31 cmn_hns_s_msmon_cfg_mbwu_flt on page 872
16'h1828	cmn_hns_s_msmon_cfg_mbwu_ctl	RW	4.3.12.32 cmn_hns_s_msmon_cfg_mbwu_ctl on page 873
16'h1840	cmn_hns_s_msmon_csu	RW	4.3.12.33 cmn_hns_s_msmon_csu on page 875
16'h1848	cmn_hns_s_msmon_csu_capture	RW	4.3.12.34 cmn_hns_s_msmon_csu_capture on page 876
16'h1860	cmn_hns_s_msmon_mbwu	RW	4.3.12.35 cmn_hns_s_msmon_mbwu on page 877
16'h1868	cmn_hns_s_msmon_mbwu_capture	RW	4.3.12.36 cmn_hns_s_msmon_mbwu_capture on page 878
16'h2000	cmn_hns_s_mpamcfg_cpbm	RW	4.3.12.37 cmn_hns_s_mpamcfg_cpbm on page 879

4.2.12 HN-I register summary

This section lists the HN-I registers used in CMN-700.

HN-I register summary

The following table shows the *HN-I* registers in offset order from the base memory address

Table 4-13: HN-I register summary

Offset	Name	Type	Description
16'h0	por_hni_node_info	RO	4.3.9.1 por_hni_node_info on page 570
16'h80	por_hni_child_info	RO	4.3.9.2 por_hni_child_info on page 570

Offset	Name	Type	Description
16'h980	por_hni_secure_register_groups_override	RW	4.3.9.3 por_hni_secure_register_groups_override on page 571
16'h900	por_hni_unit_info	RO	4.3.9.4 por_hni_unit_info on page 572
16'h908	por_hni_unit_info_1	RO	4.3.9.5 por_hni_unit_info_1 on page 574
16'hC00	por_hni_sam_addrregion0_cfg	RW	4.3.9.6 por_hni_sam_addrregion0_cfg on page 575
16'hC08	por_hni_sam_addrregion1_cfg	RW	4.3.9.7 por_hni_sam_addrregion1_cfg on page 576
16'hC10	por_hni_sam_addrregion2_cfg	RW	4.3.9.8 por_hni_sam_addrregion2_cfg on page 578
16'hC18	por_hni_sam_addrregion3_cfg	RW	4.3.9.9 por_hni_sam_addrregion3_cfg on page 579
16'hA00	por_hni_cfg_ctl	RW	4.3.9.10 por_hni_cfg_ctl on page 581
16'hA08	por_hni_aux_ctl	RW	4.3.9.11 por_hni_aux_ctl on page 582
16'h3000	por_hni_errfr	RO	4.3.9.12 por_hni_errfr on page 583
16'h3008	por_hni_errctlr	RW	4.3.9.13 por_hni_errctlr on page 584
16'h3010	por_hni_errstatus	W1C	4.3.9.14 por_hni_errstatus on page 585
16'h3018	por_hni_erraddr	RW	4.3.9.15 por_hni_erraddr on page 587
16'h3020	por_hni_errmisc	RW	4.3.9.16 por_hni_errmisc on page 588
16'h3100	por_hni_errfr_NS	RO	4.3.9.17 por_hni_errfr_NS on page 590
16'h3108	por_hni_errctlr_NS	RW	4.3.9.18 por_hni_errctlr_NS on page 591
16'h3110	por_hni_errstatus_NS	W1C	4.3.9.19 por_hni_errstatus_NS on page 592
16'h3118	por_hni_erraddr_NS	RW	4.3.9.20 por_hni_erraddr_NS on page 594
16'h3120	por_hni_errmisc_NS	RW	4.3.9.21 por_hni_errmisc_NS on page 595
16'h2000	por_hni_pmu_event_sel	RW	4.3.9.22 por_hni_pmu_event_sel on page 596
16'h2008	por_hnp_pmu_event_sel	RW	4.3.9.23 por_hnp_pmu_event_sel on page 598

4.2.13 MXP register summary

This section lists the MXP registers used in CMN-700.

MXP register summary

The following table shows the *MXP* registers in offset order from the base memory address

Table 4-14: MXP register summary

Offset	Name	Type	Description
16'h0	por_mxp_node_info	RO	4.3.13.1 por_mxp_node_info on page 880
16'h8 : 16'h30	por_mxp_device_port_connect_info_p0-5	RO	4.3.13.2 por_mxp_device_port_connect_info_p0-5 on page 881
16'h38	por_mxp_mesh_port_connect_info_east	RO	4.3.13.3 por_mxp_mesh_port_connect_info_east on page 883
16'h40	por_mxp_mesh_port_connect_info_north	RO	4.3.13.4 por_mxp_mesh_port_connect_info_north on page 884

Offset	Name	Type	Description
16'h48 : 16'h70	por_mxp_device_port_connect_ldid_info_p0-5	RO	4.3.13.5 por_mxp_device_port_connect_ldid_info_p0-5 on page 885
16'h80	por_mxp_child_info	RO	4.3.13.6 por_mxp_child_info on page 886
16'h100 : 16'h1F8	por_mxp_child_pointer_0-31	RO	4.3.13.7 por_mxp_child_pointer_0-31 on page 887
16'h900 + #{16*index}	por_mxp_p0-5_info	RO	4.3.13.8 por_mxp_p0-5_info on page 888
16'h908 + #{16*index}	por_mxp_p0-5_info_1	RO	4.3.13.9 por_mxp_p0-5_info_1 on page 889
16'h960	por_dtm_unit_info	RO	4.3.13.10 por_dtm_unit_info on page 891
16'h968 : 16'h978	por_dtm_unit_info_dt1-3	RO	4.3.13.11 por_dtm_unit_info_dt1-3 on page 891
16'h980	por_mxp_secure_register_groups_override	RW	4.3.13.12 por_mxp_secure_register_groups_override on page 892
16'hA00	por_mxp_aux_ctl	RW	4.3.13.13 por_mxp_aux_ctl on page 894
16'hA08	por_mxp_device_port_ctl	RW	4.3.13.14 por_mxp_device_port_ctl on page 894
16'hA10 : 16'hA38	por_mxp_p0-5_mpam_override	RW	4.3.13.15 por_mxp_p0-5_mpam_override on page 896
16'hA40 : 16'hA68	por_mxp_p0-5_ldid_override	RW	4.3.13.16 por_mxp_p0-5_ldid_override on page 897
16'hA80 + #{32*index}	por_mxp_p0-5_qos_control	RW	4.3.13.17 por_mxp_p0-5_qos_control on page 899
16'hA88 + #{32*index}	por_mxp_p0-5_qos_lat_tgt	RW	4.3.13.18 por_mxp_p0-5_qos_lat_tgt on page 900
16'hA90 + #{32*index}	por_mxp_p0-5_qos_lat_scale	RW	4.3.13.19 por_mxp_p0-5_qos_lat_scale on page 901
16'hA98 + #{32*index}	por_mxp_p0-5_qos_lat_range	RW	4.3.13.20 por_mxp_p0-5_qos_lat_range on page 902
16'h2000	por_mxp_pmu_event_sel	RW	4.3.13.21 por_mxp_pmu_event_sel on page 903
16'h3000	por_mxp_errfr	RO	4.3.13.22 por_mxp_errfr on page 905
16'h3008	por_mxp_errctlr	RW	4.3.13.23 por_mxp_errctlr on page 906
16'h3010	por_mxp_errstatus	W1C	4.3.13.24 por_mxp_errstatus on page 907
16'h3028	por_mxp_errmisc	RW	4.3.13.25 por_mxp_errmisc on page 909
16'h3030 : 16'h3058	por_mxp_p0-5_byte_par_err_inj	WO	4.3.13.26 por_mxp_p0-5_byte_par_err_inj on page 911
16'h3100	por_mxp_errfr_NS	RO	4.3.13.27 por_mxp_errfr_NS on page 912
16'h3108	por_mxp_errctlr_NS	RW	4.3.13.28 por_mxp_errctlr_NS on page 913
16'h3110	por_mxp_errstatus_NS	W1C	4.3.13.29 por_mxp_errstatus_NS on page 914
16'h3128	por_mxp_errmisc_NS	RW	4.3.13.30 por_mxp_errmisc_NS on page 916
16'h1C00 + #{16*index}	por_mxp_p0-5_syscoreq_ctl	RW	4.3.13.31 por_mxp_p0-5_syscoreq_ctl on page 918
16'h1C08 + #{16*index}	por_mxp_p0-5_syscoack_status	RO	4.3.13.32 por_mxp_p0-5_syscoack_status on page 919

Offset	Name	Type	Description
16'h2100	por_dtm_control	RW	4.3.13.33 por_dtm_control on page 920
16'h2118	por_dtm_fifo_entry_ready	W1C	4.3.13.34 por_dtm_fifo_entry_ready on page 921
16'h2120 + #{24*index}	por_dtm_fifo_entry0-3_0	RO	4.3.13.35 por_dtm_fifo_entry0-3_0 on page 922
16'h2128 + #{24*index}	por_dtm_fifo_entry0-3_1	RO	4.3.13.36 por_dtm_fifo_entry0-3_1 on page 923
16'h2130 + #{24*index}	por_dtm_fifo_entry0-3_2	RO	4.3.13.37 por_dtm_fifo_entry0-3_2 on page 924
16'h21A0 + #{24*index}	por_dtm_wp0-3_config	RW	4.3.13.38 por_dtm_wp0-3_config on page 925
16'h21A8 + #{24*index}	por_dtm_wp0-3_val	RW	4.3.13.39 por_dtm_wp0-3_val on page 926
16'h21B0 + #{24*index}	por_dtm_wp0-3_mask	RW	4.3.13.40 por_dtm_wp0-3_mask on page 927
16'h2200	por_dtm_pmsicr	RW	4.3.13.41 por_dtm_pmsicr on page 928
16'h2208	por_dtm_pmsirr	RW	4.3.13.42 por_dtm_pmsirr on page 929
16'h2210	por_dtm_pmu_config	RW	4.3.13.43 por_dtm_pmu_config on page 930
16'h2220	por_dtm_pmevcnt	RW	4.3.13.44 por_dtm_pmevcnt on page 934
16'h2240	por_dtm_pmevcntsr	RW	4.3.13.45 por_dtm_pmevcntsr on page 935
16'h2100 + #{512*index}	por_dtm_control_dt1-3	RW	4.3.13.46 por_dtm_control_dt1-3 on page 935
16'h2118 + #{512*index}	por_dtm_fifo_entry_ready_dt1-3	W1C	4.3.13.47 por_dtm_fifo_entry_ready_dt1-3 on page 937
16'h2120 + #{24*(index%4)} + #{512*((index/4)+1)}	por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1	RO	4.3.13.48 por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1 on page 938
16'h2128 + #{24*(index%4)} + #{512*((index/4)+1)}	por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1	RO	4.3.13.49 por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1 on page 938
16'h2130 + #{24*(index%4)} + #{512*((index/4)+1)}	por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1	RO	4.3.13.50 por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1 on page 939
16'h21A0 + #{24*(index%4)} + #{512*((index/4)+1)}	por_dtm_wp0-11%4_config_dt(0-11/4)+1	RW	4.3.13.51 por_dtm_wp0-11%4_config_dt(0-11/4)+1 on page 940
16'h21A8 + #{24*(index%4)} + #{512*((index/4)+1)}	por_dtm_wp0-11%4_val_dt(0-11/4)+1	RW	4.3.13.52 por_dtm_wp0-11%4_val_dt(0-11/4)+1 on page 941
16'h21B0 + #{24*(index%4)} + #{512*((index/4)+1)}	por_dtm_wp0-11%4_mask_dt(0-11/4)+1	RW	4.3.13.53 por_dtm_wp0-11%4_mask_dt(0-11/4)+1 on page 942
16'h2200 + #{512*index}	por_dtm_pmsicr_dt1-3	RW	4.3.13.54 por_dtm_pmsicr_dt1-3 on page 943
16'h2208 + #{512*index}	por_dtm_pmsirr_dt1-3	RW	4.3.13.55 por_dtm_pmsirr_dt1-3 on page 944
16'h2210 + #{512*index}	por_dtm_pmu_config_dt1-3	RW	4.3.13.56 por_dtm_pmu_config_dt1-3 on page 945

Offset	Name	Type	Description
16'h2220 + #{512*index}	por_dtm_pmevcnt_dt1-3	RW	4.3.13.57 por_dtm_pmevcnt_dt1-3 on page 948
16'h2240 + #{512*index}	por_dtm_pmevcntsr_dt1-3	RW	4.3.13.58 por_dtm_pmevcntsr_dt1-3 on page 949
16'hC00 : 16'hC78	por_mxp_multi_mesh_chn_sel_0-15	RW	4.3.13.59 por_mxp_multi_mesh_chn_sel_0-15 on page 950
16'hC80	por_mxp_multi_mesh_chn_ctrl	RW	4.3.13.60 por_mxp_multi_mesh_chn_ctrl on page 953
16'hC90 : 16'hCC8	por_mxp_xy_override_sel_0-7	RW	4.3.13.61 por_mxp_xy_override_sel_0-7 on page 954
16'hCD0 + #{32*index}	por_mxp_p0-5_pa2setaddr_slc	RW	4.3.13.62 por_mxp_p0-5_pa2setaddr_slc on page 956
16'hCD8 + #{32*index}	por_mxp_p0-5_pa2setaddr_sf	RW	4.3.13.63 por_mxp_p0-5_pa2setaddr_sf on page 960
16'hCE0 + #{32*index}	por_mxp_p0-5_pa2setaddr_flex_slc	RW	4.3.13.64 por_mxp_p0-5_pa2setaddr_flex_slc on page 964
16'hCE8 + #{32*index}	por_mxp_p0-5_pa2setaddr_flex_sf	RW	4.3.13.65 por_mxp_p0-5_pa2setaddr_flex_sf on page 965

4.2.14 RN-D register summary

This section lists the RN-D registers used in CMN-700.

RN-D register summary

The following table shows the *RND* registers in offset order from the base memory address

Table 4-15: RN-D register summary

Offset	Name	Type	Description
16'h0	por_rnd_node_info	RO	4.3.14.1 por_rnd_node_info on page 966
16'h80	por_rnd_child_info	RO	4.3.14.2 por_rnd_child_info on page 967
16'h980	por_rnd_secure_register_groups_override	RW	4.3.14.3 por_rnd_secure_register_groups_override on page 968
16'h900	por_rnd_unit_info	RO	4.3.14.4 por_rnd_unit_info on page 969
16'h908	por_rnd_unit_info2	RO	4.3.14.5 por_rnd_unit_info2 on page 970
16'hA00	por_rnd_cfg_ctl	RW	4.3.14.6 por_rnd_cfg_ctl on page 971
16'hA08	por_rnd_aux_ctl	RW	4.3.14.7 por_rnd_aux_ctl on page 974
16'hA10 : 16'hA20	por_rnd_s0-2_port_control	RW	4.3.14.8 por_rnd_s0-2_port_control on page 977
16'hA28 : 16'hA38	por_rnd_s0-2_mpam_control	RW	4.3.14.9 por_rnd_s0-2_mpam_control on page 979
16'hA80 + #{index*32}	por_rnd_s0-2_qos_control	RW	4.3.14.10 por_rnd_s0-2_qos_control on page 980
16'hA88 + #{index*32}	por_rnd_s0-2_qos_lat_tgt	RW	4.3.14.11 por_rnd_s0-2_qos_lat_tgt on page 982
16'hA90 + #{index*32}	por_rnd_s0-2_qos_lat_scale	RW	4.3.14.12 por_rnd_s0-2_qos_lat_scale on page 983

Offset	Name	Type	Description
16'hA98 + #{index*32}	por_rnd_s0-2_qos_lat_range	RW	4.3.14.13 por_rnd_s0-2_qos_lat_range on page 984
16'h2000	por_rnd_pmu_event_sel	RW	4.3.14.14 por_rnd_pmu_event_sel on page 985
16'h1C00	por_rnd_syscoreq_ctl	RW	4.3.14.15 por_rnd_syscoreq_ctl on page 987
16'h1C08	por_rnd_syscoack_status	RO	4.3.14.16 por_rnd_syscoack_status on page 988

4.2.15 RN-I register summary

This section lists the RN-I registers used in CMN-700.

RN-I register summary

The following table shows the *RNI* registers in offset order from the base memory address

Table 4-16: RN-I register summary

Offset	Name	Type	Description
16'h0	por_rni_node_info	RO	4.3.15.1 por_rni_node_info on page 989
16'h80	por_rni_child_info	RO	4.3.15.2 por_rni_child_info on page 990
16'h980	por_rni_secure_register_groups_override	RW	4.3.15.3 por_rni_secure_register_groups_override on page 991
16'h900	por_rni_unit_info	RO	4.3.15.4 por_rni_unit_info on page 992
16'h908	por_rni_unit_info2	RO	4.3.15.5 por_rni_unit_info2 on page 993
16'hA00	por_rni_cfg_ctl	RW	4.3.15.6 por_rni_cfg_ctl on page 994
16'hA08	por_rni_aux_ctl	RW	4.3.15.7 por_rni_aux_ctl on page 997
16'hA10 : 16'hA20	por_rni_s0-2_port_control	RW	4.3.15.8 por_rni_s0-2_port_control on page 1000
16'hA28 : 16'hA38	por_rni_s0-2_mpam_control	RW	4.3.15.9 por_rni_s0-2_mpam_control on page 1002
16'hA80 + #{index*32}	por_rni_s0-2_qos_control	RW	4.3.15.10 por_rni_s0-2_qos_control on page 1003
16'hA88 + #{index*32}	por_rni_s0-2_qos_lat_tgt	RW	4.3.15.11 por_rni_s0-2_qos_lat_tgt on page 1005
16'hA90 + #{index*32}	por_rni_s0-2_qos_lat_scale	RW	4.3.15.12 por_rni_s0-2_qos_lat_scale on page 1006
16'hA98 + #{index*32}	por_rni_s0-2_qos_lat_range	RW	4.3.15.13 por_rni_s0-2_qos_lat_range on page 1007
16'h2000	por_rni_pmu_event_sel	RW	4.3.15.14 por_rni_pmu_event_sel on page 1008

4.2.16 RN SAM register summary

This section lists the RN SAM registers used in CMN-700.

RNSAM register summary

The following table shows the *RNSAM* registers in offset order from the base memory address

Table 4-17: RNSAM register summary

Offset	Name	Type	Description
16'h0	por_rnsam_node_info	RO	4.3.16.1 por_rnsam_node_info on page 1010
16'h80	por_rnsam_child_info	RO	4.3.16.2 por_rnsam_child_info on page 1011

Offset	Name	Type	Description
16'h980	por_rnsam_secure_register_groups_override	RW	4.3.16.3 por_rnsam_secure_register_groups_override on page 1012
16'h900	por_rnsam_unit_info	RO	4.3.16.4 por_rnsam_unit_info on page 1013
16'h908	por_rnsam_unit_info1	RO	4.3.16.5 por_rnsam_unit_info1 on page 1014
{0-23} 16'hC00 : 16'hCB8	non_hash_mem_region_reg0-63	RW	4.3.16.6 non_hash_mem_region_reg0-63 on page 1016
{24-63} 16'h20C0 : 16'h24B8			
{0-23} 16'hCC0 : 16'hD78	non_hash_mem_region_cfg2_reg0-63	RW	4.3.16.7 non_hash_mem_region_cfg2_reg0-63 on page 1018
{24-63} 16'h24C0 : 16'h28D8			
16'hD80 : 16'hDF8	non_hash_tgt_nodeid0-15	RW	4.3.16.8 non_hash_tgt_nodeid0-15 on page 1019
16'h11A0	cml_port_aggr_mode_ctrl_reg	RW	4.3.16.9 cml_port_aggr_mode_ctrl_reg on page 1020
{1-3} 16'h11A8 : 16'h11B8	cml_port_aggr_mode_ctrl_reg1-6	RW	4.3.16.10 cml_port_aggr_mode_ctrl_reg1-6 on page 1021
{4-6} 16'h2A20 : 16'h2A30			
16'hE00 : 16'hE18	sys_cache_grp_region0-3	RW	4.3.16.11 sys_cache_grp_region0-3 on page 1024
{4-7} 16'hE20 : 16'hE38	hashed_tgt_grp_cfg1_region4-31	RW	4.3.16.12 hashed_tgt_grp_cfg1_region4-31 on page 1025
{8-31} 16'h3040 : 16'h30F8			
16'h3100 : 16'h31F8	hashed_tgt_grp_cfg2_region0-31	RW	4.3.16.13 hashed_tgt_grp_cfg2_region0-31 on page 1027
16'hE40 : 16'hE58	sys_cache_grp_secondary_reg0-3	RW	4.3.16.14 sys_cache_grp_secondary_reg0-3 on page 1028
{4-7} 16'hE60 : 16'hE78	hashed_target_grp_secondary_cfg1_reg4-31	RW	4.3.16.15 hashed_target_grp_secondary_cfg1_reg4-31 on page 1030
{8-31} 16'h3240 : 16'h32F8			
16'h3300 : 16'h33F8	hashed_target_grp_secondary_cfg2_reg0-31	RW	4.3.16.16 hashed_target_grp_secondary_cfg2_reg0-31 on page 1032
16'h3400 : 16'h34F8	hashed_target_grp_hash_cntl_reg0-31	RW	4.3.16.17 hashed_target_grp_hash_cntl_reg0-31 on page 1033
16'hEA0	sys_cache_group_hn_count	RW	4.3.16.18 sys_cache_group_hn_count on page 1035

Offset	Name	Type	Description
{1} 16'hEA8	hashed_target_group_hn_count_reg1-3	RW	4.3.16.19 hashed_target_group_hn_count_reg1-3 on page 1036
{2-3} 16'h3710 : 16'h3718			
16'hEC0	sys_cache_grp_nonhash_nodeid	RW	4.3.16.20 sys_cache_grp_nonhash_nodeid on page 1037
{0-5} 16'hEC8 : 16'hEE8	hashed_target_grp_nonhash_nodeid_reg1-6	RW	4.3.16.21 hashed_target_grp_nonhash_nodeid_reg1-6 on page 1038
{6} 16'h3800			
16'hF00 : 16'hF78	sys_cache_grp_hn_nodeid_reg0-15	RW	4.3.16.22 sys_cache_grp_hn_nodeid_reg0-15 on page 1039
16'hF80 : 16'hFF8	hashed_target_grp_hnf_nodeid_reg16-31	RW	4.3.16.23 hashed_target_grp_hnf_nodeid_reg16-31 on page 1041
16'h3600 : 16'h3678	hashed_target_grp_hnp_nodeid_reg0-15	RW	4.3.16.24 hashed_target_grp_hnp_nodeid_reg0-15 on page 1042
16'h1120	sys_cache_grp_cal_mode_reg	RW	4.3.16.25 sys_cache_grp_cal_mode_reg on page 1043
{1-3} 16'h1128 : 16'h1138	hashed_target_grp_cal_mode_reg1-7	RW	4.3.16.26 hashed_target_grp_cal_mode_reg1-7 on page 1045
{4-7} 16'h37A0 : 16'h37B8			
16'h1180	sys_cache_grp_hn_cpa_en_reg	RW	4.3.16.27 sys_cache_grp_hn_cpa_en_reg on page 1047
16'h1188	hashed_target_grp_hnf_cpa_en_reg1-1	RW	4.3.16.28 hashed_target_grp_hnf_cpa_en_reg1-1 on page 1048
16'h3900 : 16'h3978	hashed_target_grp_cpag_perhnf_reg0-15	RW	4.3.16.29 hashed_target_grp_cpag_perhnf_reg0-15 on page 1049
16'h1190	sys_cache_grp_hn_cpa_grp_reg	RW	4.3.16.30 sys_cache_grp_hn_cpa_grp_reg on page 1050
{1} 16'h1198	hashed_target_grp_cpa_grp_reg1-7	RW	4.3.16.31 hashed_target_grp_cpa_grp_reg1-7 on page 1052
{2-7} 16'h3750 : 16'h3778			
16'h37C0 : 16'h37C8	hashed_target_grp_hnf_lcn_bound_cfg_reg0-1	RW	4.3.16.32 hashed_target_grp_hnf_lcn_bound_cfg_reg0-1 on page 1054
16'h37E0 : 16'h37E8	hashed_target_grp_hnf_target_type_override_cfg_reg0-1	RW	4.3.16.33 hashed_target_grp_hnf_target_type_override_cfg_reg0-1 on page 1054
16'h3A00 : 16'h3AF8	hashed_target_grp_compact_cpag_ctrl0-31	RW	4.3.16.34 hashed_target_grp_compact_cpag_ctrl0-31 on page 1055
16'h3B00 : 16'h3BF8	hashed_target_grp_compact_hash_ctrl0-31	RW	4.3.16.35 hashed_target_grp_compact_hash_ctrl0-31 on page 1057
16'hE80	rnsam_hash_addr_mask_reg	RW	4.3.16.36 rnsam_hash_addr_mask_reg on page 1061
16'hE88	rnsam_hash_axi_id_mask_reg	RW	4.3.16.37 rnsam_hash_axi_id_mask_reg on page 1062
16'hE90	rnsam_region_cmp_addr_mask_reg	RW	4.3.16.38 rnsam_region_cmp_addr_mask_reg on page 1063

Offset	Name	Type	Description
{0-5} 16'h11C0 : 16'h11E8	cml_port_aggr_grp0-31_add_mask	RW	4.3.16.39 cml_port_aggr_grp0-31_add_mask on page 1063
{6-31} 16'h2B30 : 16'h2BF8			
16'h2B00 : 16'h2B18	cml_cpag_base_indx_grp0-3	RW	4.3.16.40 cml_cpag_base_indx_grp0-3 on page 1064
{0-2} 16'h11F0 : 16'h1200	cml_port_aggr_grp_reg0-12	RW	4.3.16.41 cml_port_aggr_grp_reg0-12 on page 1066
{3-12} 16'h2C18 : 16'h2C60			
16'h1208	cml_port_aggr_ctrl_reg	RW	4.3.16.42 cml_port_aggr_ctrl_reg on page 1067
16'h1210 : 16'h1238	cml_port_aggr_ctrl_reg1-6	RW	4.3.16.43 cml_port_aggr_ctrl_reg1-6 on page 1070
16'hEB0	sys_cache_grp_sn_attr	RW	4.3.16.44 sys_cache_grp_sn_attr on page 1073
16'hEB8	sys_cache_grp_sn_attr1	RW	4.3.16.45 sys_cache_grp_sn_attr1 on page 1076
16'h1140 : 16'h1158	sys_cache_grp_sn_sam_cfg0-3	RW	4.3.16.46 sys_cache_grp_sn_sam_cfg0-3 on page 1079
16'h1280 : 16'h12F8	sam_qos_mem_region_reg0-15	RW	4.3.16.47 sam_qos_mem_region_reg0-15 on page 1080
16'h1340 : 16'h13B8	sam_qos_mem_region_cfg2_reg0-15	RW	4.3.16.48 sam_qos_mem_region_cfg2_reg0-15 on page 1082
16'h4000 : 16'h4FF8	sam_scg0-511/64_prefetch_nonhashed_ mem_region_cfg1_reg0-511%64	RW	4.3.16.49 sam_scg0-511/64_prefetch_nonhashed_ mem_region_cfg1_reg0-511%64 on page 1083
16'h5000 : 16'h5FF8	sam_scg0-511/64_prefetch_nonhashed_ mem_region_cfg2_reg0-511%64	RW	4.3.16.50 sam_scg0-511/64_prefetch_nonhashed_ mem_region_cfg2_reg0-511%64 on page 1084
16'h6000 : 16'h61F8	sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8	RW	4.3.16.51 sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8 on page 1086
16'h6200 : 16'h63F8	sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8	RW	4.3.16.52 sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8 on page 1087
16'h6400 : 16'h65F8	sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8	RW	4.3.16.53 sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8 on page 1088
16'h1000 : 16'h10F8	sys_cache_grp_sn_nodeid_reg0-31	RW	4.3.16.54 sys_cache_grp_sn_nodeid_reg0-31 on page 1090
16'h1400 : 16'h15F8	sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32	RW	4.3.16.55 sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32 on page 1091
16'h6600 : 16'h6678	sys_cache_grp_hashed_regions_sn_nodeid_reg0-15	RW	4.3.16.56 sys_cache_grp_hashed_regions_sn_nodeid_reg0-15 on page 1093
16'h1100	rnsam_status	RW	4.3.16.57 rnsam_status on page 1094
16'h1108	gic_mem_region_reg	RW	4.3.16.58 gic_mem_region_reg on page 1095

Offset	Name	Type	Description
16'h1600 : 16'h1638	sam_generic_regs0-7	RW	4.3.16.59 sam_generic_regs0-7 on page 1097

4.2.17 SBSX register summary

This section lists the SBSX registers used in CMN-700.

SBSX register summary

The following table shows the *SBSX* registers in offset order from the base memory address

Table 4-18: SBSX register summary

Offset	Name	Type	Description
16'h0	por_sbsx_node_info	RO	4.3.17.1 por_sbsx_node_info on page 1098
16'h80	por_sbsx_child_info	RO	4.3.17.2 por_sbsx_child_info on page 1099
16'h980	por_sbsx_secure_register_groups_override	RW	4.3.17.3 por_sbsx_secure_register_groups_override on page 1100
16'h900	por_sbsx_unit_info	RO	4.3.17.4 por_sbsx_unit_info on page 1101
16'hA00	por_sbsx_cfg_ctl	RW	4.3.17.5 por_sbsx_cfg_ctl on page 1102
16'hA08	por_sbsx_aux_ctl	RW	4.3.17.6 por_sbsx_aux_ctl on page 1104
16'hA18	por_sbsx_cbusy_limit_ctl	RW	4.3.17.7 por_sbsx_cbusy_limit_ctl on page 1105
16'h3000	por_sbsx_errfr	RO	4.3.17.8 por_sbsx_errfr on page 1106
16'h3008	por_sbsx_errctlr	RW	4.3.17.9 por_sbsx_errctlr on page 1107
16'h3010	por_sbsx_errstatus	W1C	4.3.17.10 por_sbsx_errstatus on page 1108
16'h3018	por_sbsx_erraddr	RW	4.3.17.11 por_sbsx_erraddr on page 1110
16'h3020	por_sbsx_errmisc	RW	4.3.17.12 por_sbsx_errmisc on page 1111
16'h3100	por_sbsx_errfr_NS	RO	4.3.17.13 por_sbsx_errfr_NS on page 1112
16'h3108	por_sbsx_errctlr_NS	RW	4.3.17.14 por_sbsx_errctlr_NS on page 1113
16'h3110	por_sbsx_errstatus_NS	W1C	4.3.17.15 por_sbsx_errstatus_NS on page 1114
16'h3118	por_sbsx_erraddr_NS	RW	4.3.17.16 por_sbsx_erraddr_NS on page 1116
16'h3120	por_sbsx_errmisc_NS	RW	4.3.17.17 por_sbsx_errmisc_NS on page 1117
16'h2000	por_sbsx_pmu_event_sel	RW	4.3.17.18 por_sbsx_pmu_event_sel on page 1118

4.3 Register descriptions

This section contains register descriptions.

4.3.1 APB register descriptions

This section lists the APB registers.

4.3.1.1 por_apb_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-3: por_apb_node_info

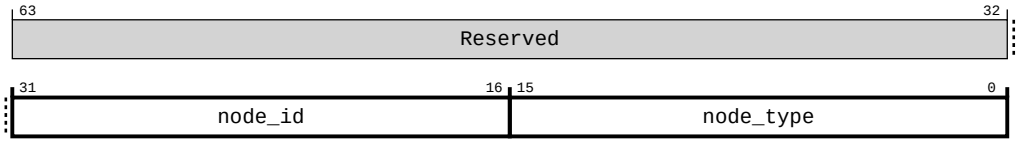


Table 4-19: por_apb_node_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h1000

4.3.1.2 por_apb_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-4: por_apb_child_info

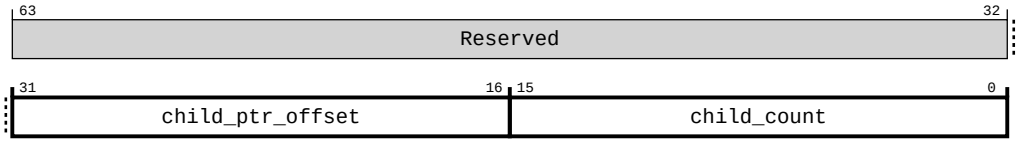


Table 4-20: por_apb_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0000
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.1.3 por_apb_only_access

Functions as the CMN access control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-5: por_apb_only_access

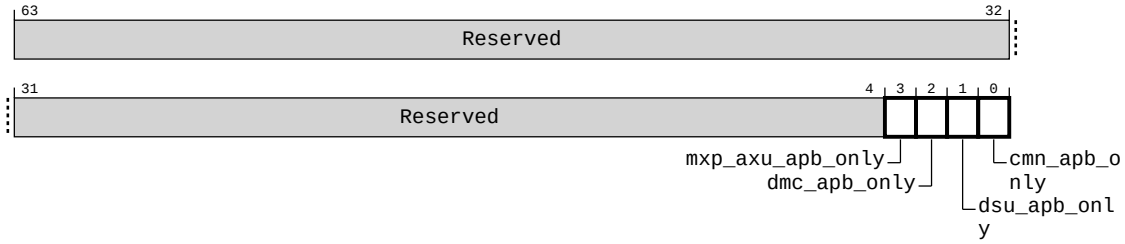


Table 4-21: por_apb_only_access attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	mxp_axu_apb_only	MXP AXI utility bus access by APB only **1'b0** MXP AXU accessible via CHI/AXI in addition to APB **1'b1** MXP AXU accessible only via APB	RW	1'b0
[2]	dmc_apb_only	DMC AXI utility bus access by APB only **1'b0** DMC AXU accessible via CHI/AXI in addition to APB **1'b1** DMC AXU accessible only via APB	RW	1'b0
[1]	dsu_apb_only	DSU AXI utility bus access by APB only **1'b0** DSU AXU accessible via CHI/AXI in addition to APB **1'b1** DSU AXU accessible only via APB	RW	1'b0
[0]	cmn_apb_only	CMN config access by APB only **1'b0** CMN config accessible via CHI/AXI in addition to APB **1'b1** CMN config accessible only via APB	RW	1'b0

4.3.1.4 por_axu_control

Functions as the CMN AXU interface control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h988`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-6: por_axu_control

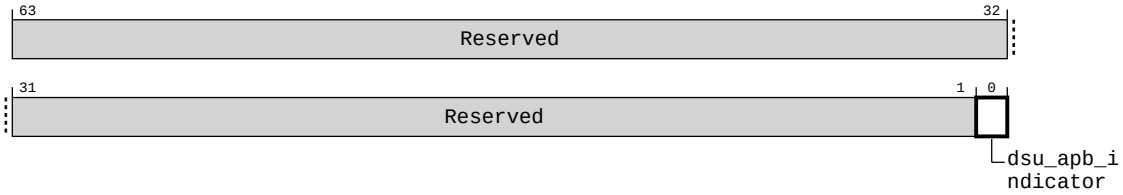


Table 4-22: por_axu_control attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	dsu_apb_indicator	DSU AxADDRU[23] access indicator enable **1'b1** AxADDRU[23] will carry source indicator (0: from CHI / 1: from APB) **1'b0** AxADDRU[23] behaves as default	RW	1'b0

4.3.2 CCG_HA register descriptions

This section lists the CCG_HA registers.

4.3.2.1 por_ccg_ha_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-7: por_ccg_ha_node_info

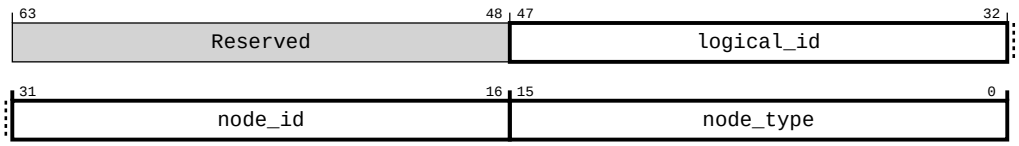


Table 4-23: por_ccg_ha_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0104

4.3.2.2 por_ccg_ha_id

Contains the CCIX-assigned HAID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h8`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-8: por_ccg_ha_id

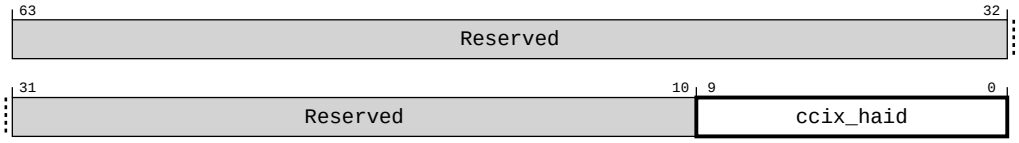


Table 4-24: por_ccg_ha_id attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:0]	ccix_haid	CCIX HAID	RW	10'h0

4.3.2.3 por_ccg_ha_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-9: por_ccg_ha_child_info

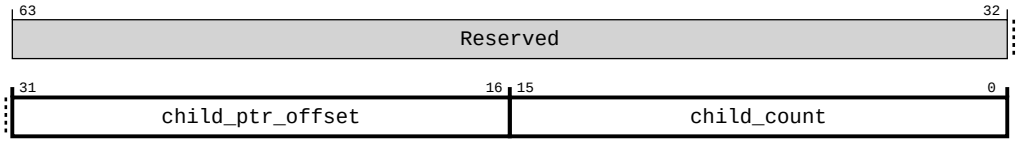


Table 4-25: por_ccg_ha_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'h0

4.3.2.4 por_ccg_ha_cfg_ctl

Functions as the configuration control register. Specifies the current mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-10: por_ccg_ha_cfg_ctl

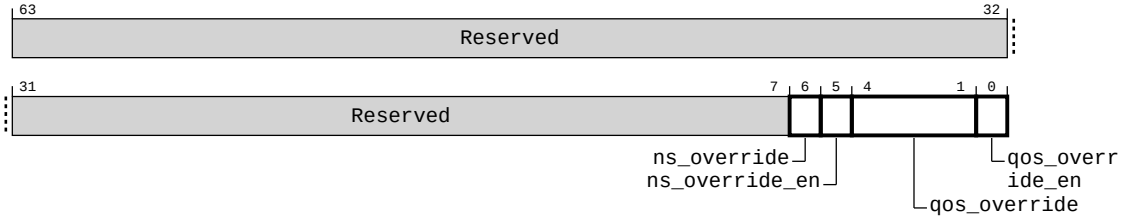


Table 4-26: por_ccg_ha_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	ns_override	NS override value	RW	1'b0
[5]	ns_override_en	NS override en When set, NS(Non-secure) value on CHI side is driven from NS override value in this register	RW	1'b0
[4:1]	qos_override	QoS override value	RW	4'b0
[0]	qos_override_en	QoS override en When set, QoS value on CHI side is driven from QoS override value in this register	RW	1'b0

4.3.2.5 por_ccg_ha_aux_ctl

Functions as the auxiliary control register for CXHA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA08

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-11: por_ccg_ha_aux_ctl

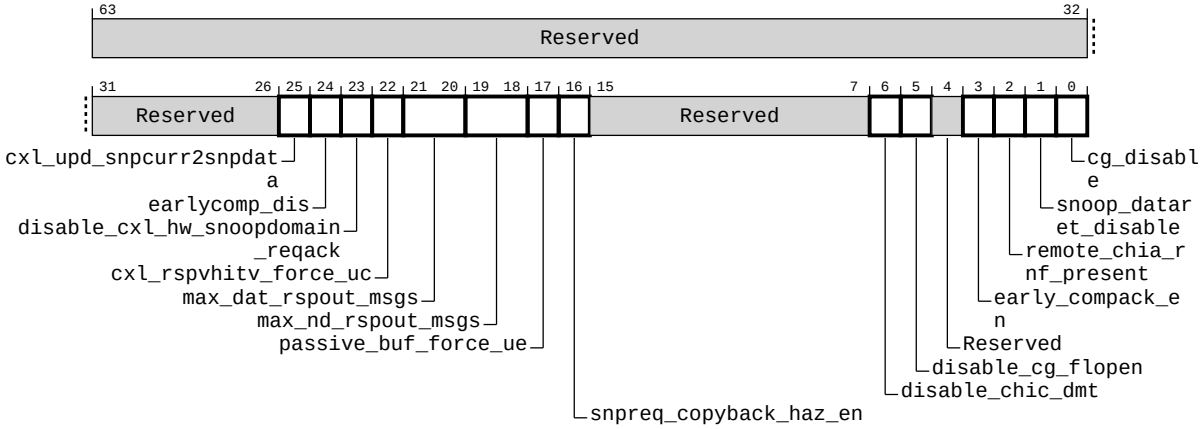


Table 4-27: por_ccg_ha_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	cxl_upd_snpcurr2snpdata	When set, updates SnpCurr to SnpData on CXL.Cache	RW	1'b0
[24]	earlycomp_dis	When set, disables sending early completions for requests	RW	1'b0
[23]	disable_cxl_hw_snoopdomain_reqack	When set, disables CXL Hardware based Snoop domain handshake	RW	1'b0
[22]	cxl_rspvhitv_force_uc	When set, forces UC response state for Snoop response on CHI when D2H RspVHitV response is received on CXL	RW	1'b0

Bits	Name	Description	Type	Reset
[21:20]	max_dat_rspout_msgs	Used to configure the maximum number of response data messages (Read Data) presented to CCLA's packing logic. **2'b00** one message **2'b01** two messages **2'b10** three messages **2'b11** four messages Note: The max is further limited by max allowed by the given protocol (CCIX2.0/CXL.mem/CXL.cache)	RW	2'b11
[19:18]	max_nd_rspout_msgs	Used to configure the maximum number of non-data response messages (Completions/GO/WritePulls) presented to CCLA's packing logic. **2'b00** one message **2'b01** two messages **2'b10** three messages **2'b11** four messages Note: The max is further limited by max allowed by the given protocol (CCIX2.0/CXL.mem/CXL.cache)	RW	2'b11
[17]	passive_buf_force_ue	When set, focres all DE's originating in Passive Buffer to be reported as UEs	RW	1'b0
[16]	snpreq_copyback_haz_en	When set enables Snoop-CopyBack hazarding at incoming Snoop Request pipe	RW	1'b0
[15:7]	Reserved	Reserved	RO	-
[6]	disable_chic_dmt	When set disables CHI-C style DMT	RW	1'b0
[5]	disable_cg_flopen	Disables enhanced flop enable control for dynamic power savings	RW	1'b0
[4]	Reserved	Reserved	RO	-
[3]	early_compack_en	Early CompAck enable; enables sending early CompAck on CCIX for requests that require CompAck	RW	1'b1
[2]	remote_chia_rnf_present	Indicates existence of CHIA RN-F in system; HA uses this indication to send SnpToS or SnpToSC **1'b0** HA converts SnpShared, SnpClean, and SnpNotSharedDirty to SnpToSC **1'b1** HA converts SnpShared, SnpClean, and SnpNotSharedDirty to SnpToS	RW	1'b0
[1]	snoop_dataret_disable	Disables setting data return for CCIX snoop requests for all CHI snoop opcodes	RW	1'b0
[0]	cg_disable	Disables clock gating when set	RW	1'b0

4.3.2.6 por_ccg_ha_mpam_control_link0

Controls MPAM override values on incoming CCIX Request in non-SMP mode for Link0

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA10`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.mpam_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-12: por_ccg_ha_mpam_control_link0

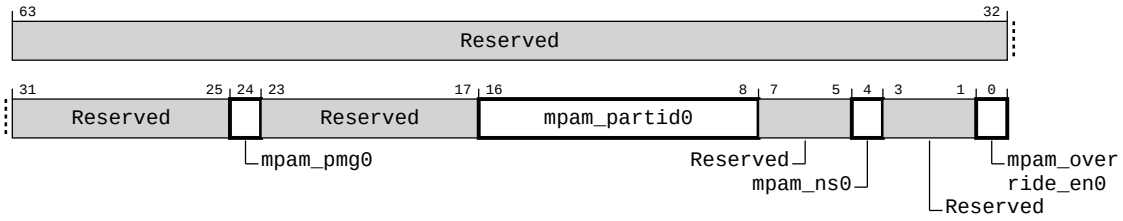


Table 4-28: por_ccg_ha_mpam_control_link0 attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	mpam_pmg0	MPAM_PMG value	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	mpam_partid0	MPAM_PARTID value	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	mpam_ns0	MPAM_NS value	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	mpam_override_en0	MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Applicable only in non-SMP mode	RW	1'b0

4.3.2.7 por_ccg_ha_mpam_control_link1

Controls MPAM override values on incoming CCIX Request in non-SMP mode for Link1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA18

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.mpam_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-13: por_ccg_ha_mpam_control_link1

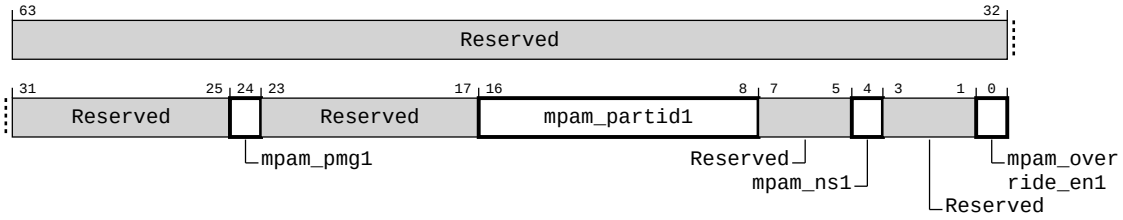


Table 4-29: por_ccg_ha_mpam_control_link1 attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	mpam_pmg1	MPAM_PMG value	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	mpam_partid1	MPAM_PARTID value	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	mpam_ns1	MPAM_NS value	RW	1'b0

Bits	Name	Description	Type	Reset
[3:1]	Reserved	Reserved	RO	-
[0]	mpam_override_en1	MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Applicable only in non-SMP mode	RW	1'b0

4.3.2.8 por_ccg_ha_mpam_control_link2

Controls MPAM override values on incoming CCIX Request in non-SMP mode for Link2

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA20`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.mpam_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-14: por_ccg_ha_mpam_control_link2

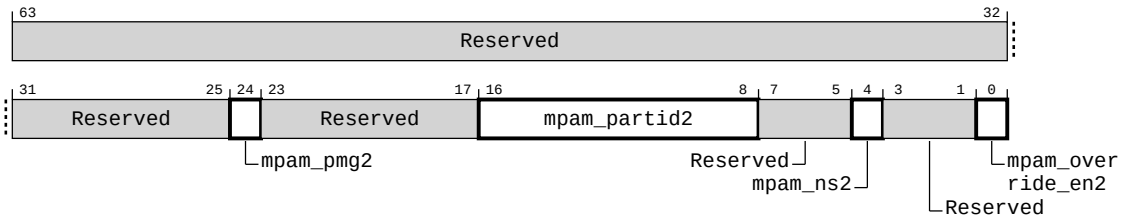


Table 4-30: por_ccg_ha_mpam_control_link2 attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	mpam_pmg2	MPAM_PMG value	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	mpam_partid2	MPAM_PARTID value	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	mpam_ns2	MPAM_NS value	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	mpam_override_en2	MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Applicable only in non-SMP mode	RW	1'b0

4.3.2.9 por_ccg_ha_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-15: por_ccg_ha_secure_register_groups_override

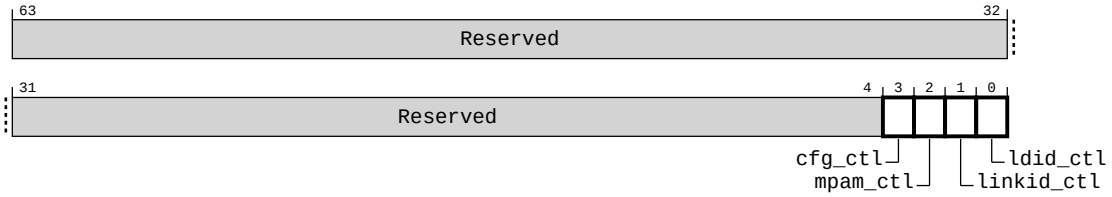


Table 4-31: por_ccg_ha_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	cfg_ctl	Allows Non-secure access to Secure HA config control registers	RW	1'b0
[2]	mpam_ctl	Allows Non-secure access to Secure HA MPAM override registers	RW	1'b0
[1]	linkid_ctl	Allows Non-secure access to Secure HA Link ID registers	RW	1'b0
[0]	ldid_ctl	Allows Non-secure access to Secure HA LDID registers	RW	1'b0

4.3.2.10 por_ccg_ha_unit_info

Provides component identification information for CXHA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-16: por_ccg_ha_unit_info

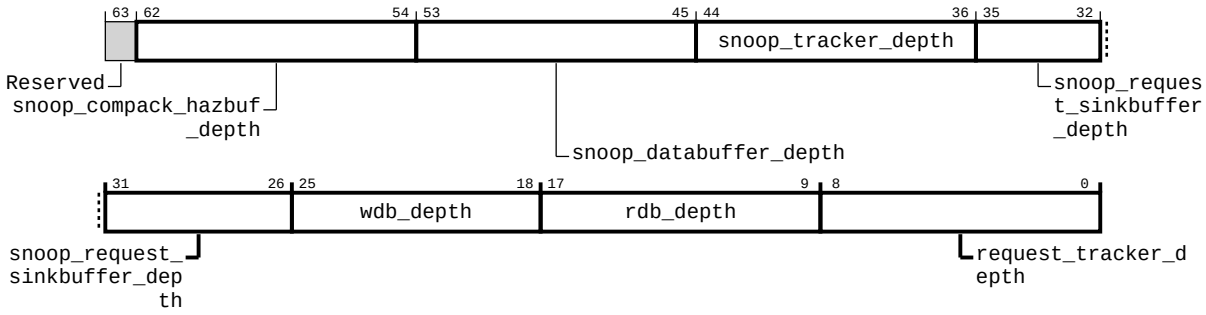


Table 4-32: por_ccg_ha_unit_info attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:54]	snoop_compack_hazbuf_depth	Depth of CompAck snoop hazard buffer	RO	Configuration dependent
[53:45]	snoop_databuffer_depth	Depth of snoop data buffer	RO	Configuration dependent
[44:36]	snoop_tracker_depth	Depth of snoop tracker; number of outstanding SNP requests on CCIX	RO	Configuration dependent
[35:26]	snoop_request_sinkbuffer_depth	Depth of snoop request sink buffer; number of CHI SNP requests that can be sunk by CXHA	RO	Configuration dependent
[25:18]	wdb_depth	Depth of write data buffer	RO	Configuration dependent
[17:9]	rdb_depth	Depth of read data buffer	RO	Configuration dependent
[8:0]	request_tracker_depth	Depth of request tracker	RO	Configuration dependent

4.3.2.11 por_ccg_ha_unit_info2

Provides additonal component identification information for CXHA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-17: por_ccg_ha_unit_info2

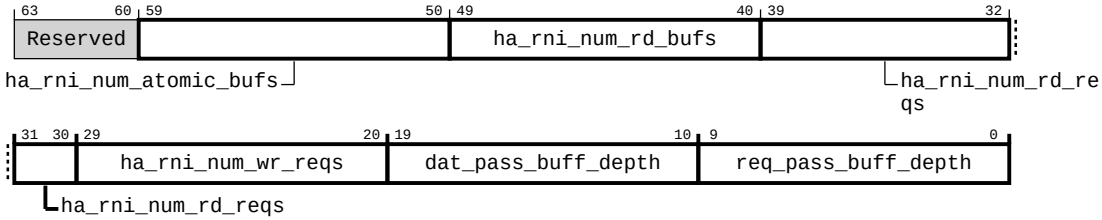


Table 4-33: por_ccg_ha_unit_info2 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:50]	ha_rni_num_atomic_bufs	Number of HA_RNI atomic data buffers	RO	Configuration dependent
[49:40]	ha_rni_num_rd_bufs	Number of HA_RNI read data buffers	RO	Configuration dependent
[39:30]	ha_rni_num_rd_reqs	Number of HA_RNI outstanding read requests	RO	Configuration dependent
[29:20]	ha_rni_num_wr_reqs	Number of HA_RNI outstanding write requests	RO	Configuration dependent
[19:10]	dat_pass_buff_depth	Depth of DAT Passive Buffer	RO	Configuration dependent
[9:0]	req_pass_buff_depth	Depth of REQ Passive Buffer	RO	Configuration dependent

4.3.2.12 por_ccg_ha_unit_info3

Provides additonal component identification information for CXHA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h910`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-18: por_ccg_ha_unit_info3

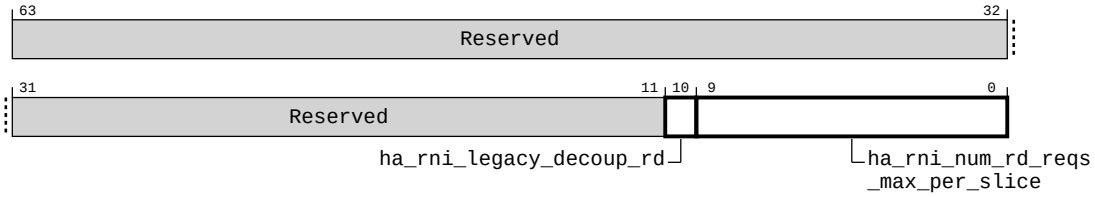


Table 4-34: por_ccg_ha_unit_info3 attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	ha_rni_legacy_decoup_rd	Legacy decoupled read mode, no read burst propagation	RO	Configuration dependent
[9:0]	ha_rni_num_rd_reqs_max_per_slice	Number of HA_RNI read request entries per slice	RO	Configuration dependent

4.3.2.13 por_ccg_ha_agentid_to_linkid_reg0

Specifies the mapping of Agent ID to Link ID for Agent IDs 0 to 7.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-19: por_ccg_ha_agentid_to_linkid_reg0

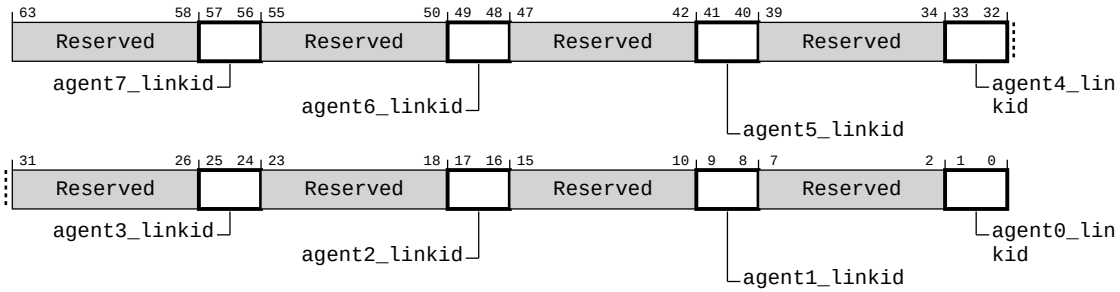


Table 4-35: por_ccg_ha_agentid_to_linkid_reg0 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent7_linkid	Specifies Link ID 7	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent6_linkid	Specifies Link ID 6	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent5_linkid	Specifies Link ID 5	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent4_linkid	Specifies Link ID 4	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent3_linkid	Specifies Link ID 3	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent2_linkid	Specifies Link ID 2	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent1_linkid	Specifies Link ID 1	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent0_linkid	Specifies Link ID 0	RW	2'h0

4.3.2.14 por_ccg_ha_agentid_to_linkid_reg1

Specifies the mapping of Agent ID to Link ID for Agent IDs 8 to 15.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F08`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-20: por_ccg_ha_agentid_to_linkid_reg1

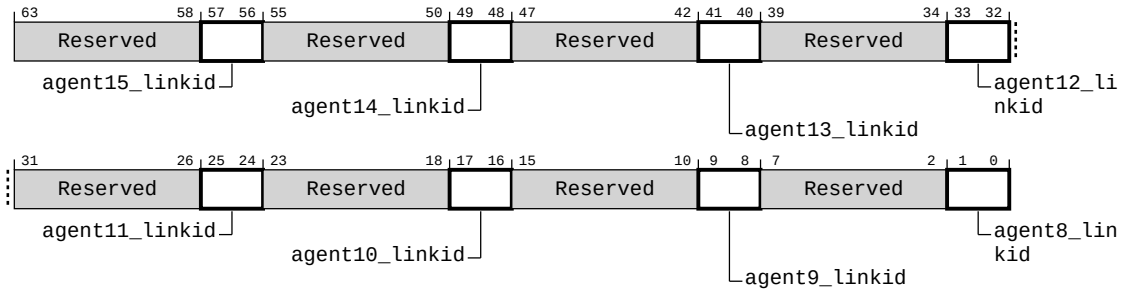


Table 4-36: por_ccg_ha_agentid_to_linkid_reg1 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent15_linkid	Specifies Link ID 15	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent14_linkid	Specifies Link ID 14	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent13_linkid	Specifies Link ID 13	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent12_linkid	Specifies Link ID 12	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent11_linkid	Specifies Link ID 11	RW	2'h0
[23:18]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[17:16]	agent10_linkid	Specifies Link ID 10	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent9_linkid	Specifies Link ID 9	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent8_linkid	Specifies Link ID 8	RW	2'h0

4.3.2.15 por_ccg_ha_agentid_to_linkid_reg2

Specifies the mapping of Agent ID to Link ID for Agent IDs 16 to 23.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F10`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-21: por_ccg_ha_agentid_to_linkid_reg2

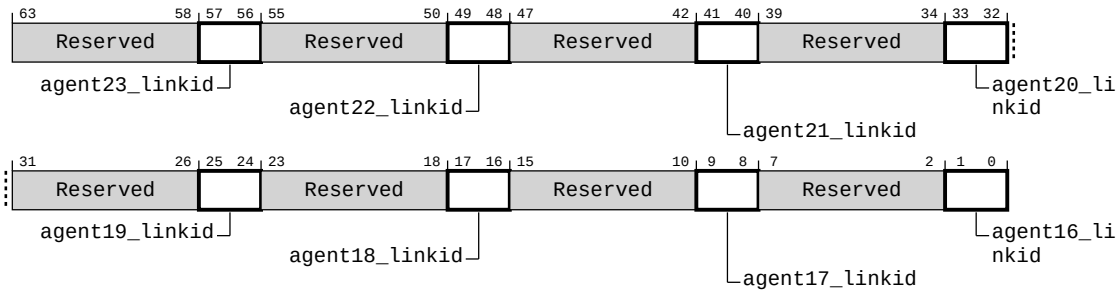


Table 4-37: por_ccg_ha_agentid_to_linkid_reg2 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent23_linkid	Specifies Link ID 23	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent22_linkid	Specifies Link ID 22	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent21_linkid	Specifies Link ID 21	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent20_linkid	Specifies Link ID 20	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent19_linkid	Specifies Link ID 19	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent18_linkid	Specifies Link ID 18	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent17_linkid	Specifies Link ID 17	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent16_linkid	Specifies Link ID 16	RW	2'h0

4.3.2.16 por_ccg_ha_agentid_to_linkid_reg3

Specifies the mapping of Agent ID to Link ID for Agent IDs 24 to 31.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F18`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-22: por_ccg_ha_agentid_to_linkid_reg3

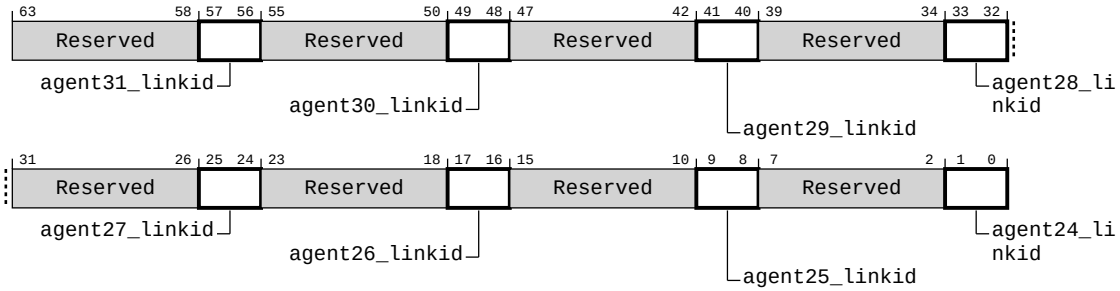


Table 4-38: por_ccg_ha_agentid_to_linkid_reg3 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent31_linkid	Specifies Link ID 31	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent30_linkid	Specifies Link ID 30	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent29_linkid	Specifies Link ID 29	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent28_linkid	Specifies Link ID 28	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent27_linkid	Specifies Link ID 27	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent26_linkid	Specifies Link ID 26	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent25_linkid	Specifies Link ID 25	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent24_linkid	Specifies Link ID 24	RW	2'h0

4.3.2.17 por_ccg_ha_agentid_to_linkid_reg4

Specifies the mapping of Agent ID to Link ID for Agent IDs 32 to 39.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F20`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-23: por_ccg_ha_agentid_to_linkid_reg4

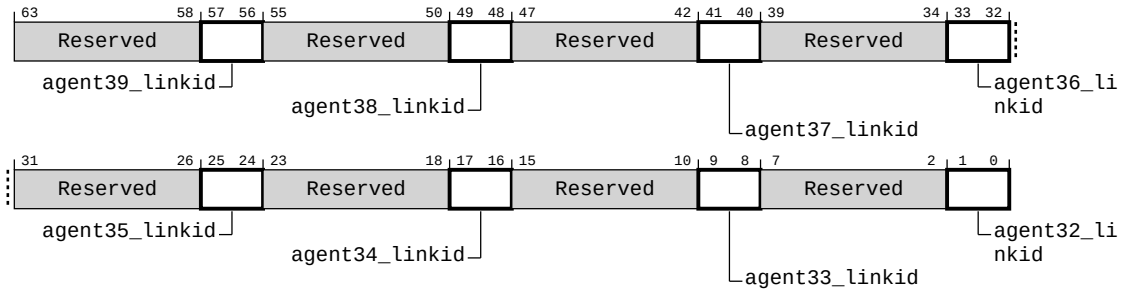


Table 4-39: por_ccg_ha_agentid_to_linkid_reg4 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent39_linkid	Specifies Link ID 39	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent38_linkid	Specifies Link ID 38	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent37_linkid	Specifies Link ID 37	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent36_linkid	Specifies Link ID 36	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent35_linkid	Specifies Link ID 35	RW	2'h0
[23:18]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[17:16]	agent34_linkid	Specifies Link ID 34	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent33_linkid	Specifies Link ID 33	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent32_linkid	Specifies Link ID 32	RW	2'h0

4.3.2.18 por_ccg_ha_agentid_to_linkid_reg5

Specifies the mapping of Agent ID to Link ID for Agent IDs 40 to 47.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F28`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-24: por_ccg_ha_agentid_to_linkid_reg5

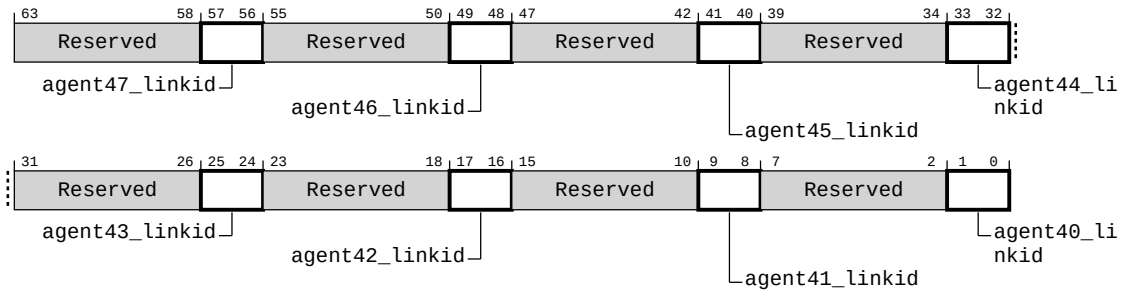


Table 4-40: por_ccg_ha_agentid_to_linkid_reg5 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent47_linkid	Specifies Link ID 47	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent46_linkid	Specifies Link ID 46	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent45_linkid	Specifies Link ID 45	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent44_linkid	Specifies Link ID 44	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent43_linkid	Specifies Link ID 43	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent42_linkid	Specifies Link ID 42	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent41_linkid	Specifies Link ID 41	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent40_linkid	Specifies Link ID 40	RW	2'h0

4.3.2.19 por_ccg_ha_agentid_to_linkid_reg6

Specifies the mapping of Agent ID to Link ID for Agent IDs 48 to 55.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F30`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-25: por_ccg_ha_agentid_to_linkid_reg6

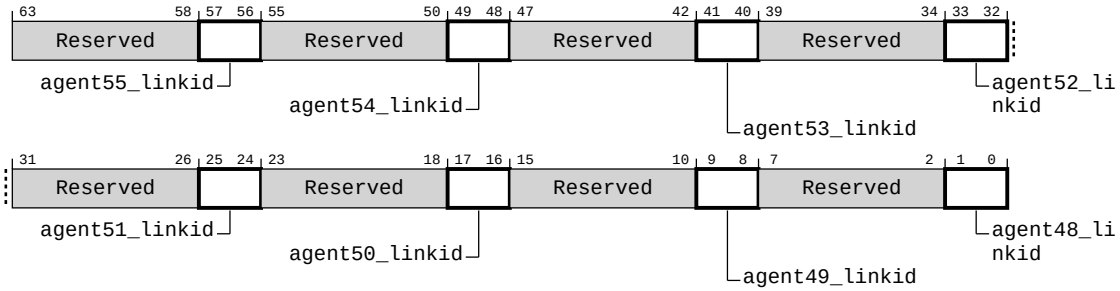


Table 4-41: por_ccg_ha_agentid_to_linkid_reg6 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent55_linkid	Specifies Link ID 55	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent54_linkid	Specifies Link ID 54	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent53_linkid	Specifies Link ID 53	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent52_linkid	Specifies Link ID 52	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent51_linkid	Specifies Link ID 51	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent50_linkid	Specifies Link ID 50	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent49_linkid	Specifies Link ID 49	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent48_linkid	Specifies Link ID 48	RW	2'h0

4.3.2.20 por_ccg_ha_agentid_to_linkid_reg7

Specifies the mapping of Agent ID to Link ID for Agent IDs 56 to 63.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1F38`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-26: por_ccg_ha_agentid_to_linkid_reg7

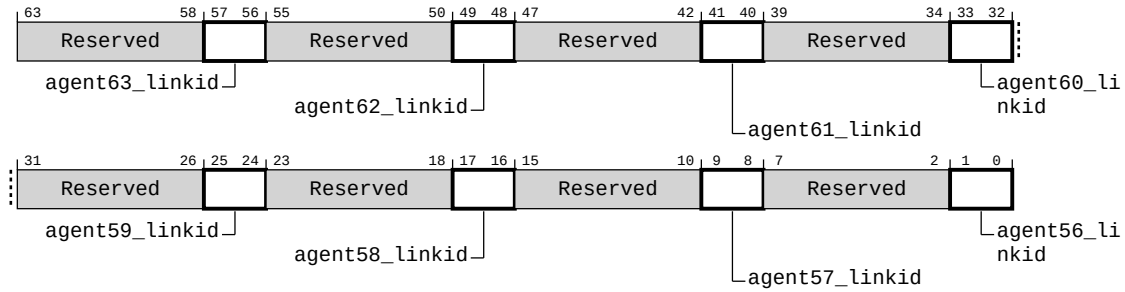


Table 4-42: por_ccg_ha_agentid_to_linkid_reg7 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent63_linkid	Specifies Link ID 63	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent62_linkid	Specifies Link ID 62	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent61_linkid	Specifies Link ID 61	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent60_linkid	Specifies Link ID 60	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent59_linkid	Specifies Link ID 59	RW	2'h0
[23:18]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[17:16]	agent58_linkid	Specifies Link ID 58	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent57_linkid	Specifies Link ID 57	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent56_linkid	Specifies Link ID 56	RW	2'h0

4.3.2.21 por_ccg_ha_agentid_to_linkid_val

Specifies which Agent ID to Link ID mappings are valid.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1FF8`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-27: por_ccg_ha_agentid_to_linkid_val

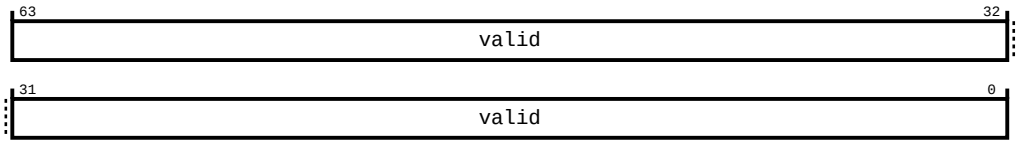


Table 4-43: por_ccg_ha_agentid_to_linkid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Link ID is valid; bit number corresponds to logical Agent ID number (from 0 to 63)	RW	63'h0

4.3.2.22 por_ccg_ha_rnf_exp_raid_to_ldid_reg_0-255

There are 256 iterations of this register. The index ranges from 0 to 255. Specifies the mapping of Expanded RAID to RN-F LDID for Expanded RAIDs #{index*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ha_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-28: por_ccg_ha_rnf_exp_raid_to_ldid_reg_0-255

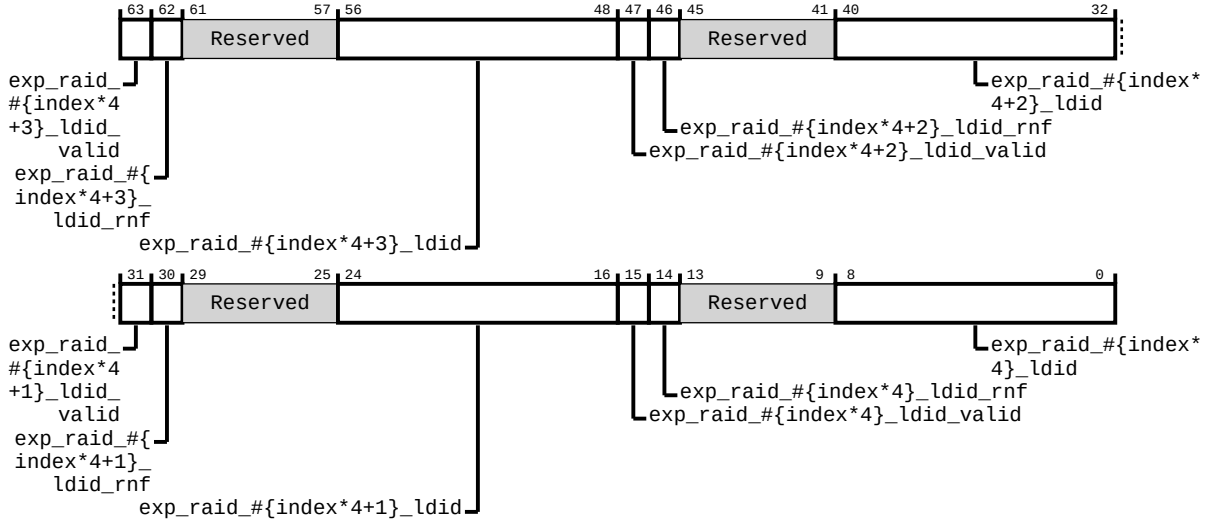


Table 4-44: por_ccg_ha_rnf_exp_raid_to_ldid_reg_0-255 attributes

Bits	Name	Description	Type	Reset
[63]	exp_raid_#{index*4+3}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+3} is valid	RW	1'b0
[62]	exp_raid_#{index*4+3}_ldid_rnf	Specifies if Expanded RAID #{index*4+3} is RN-F	RW	1'b0
[61:57]	Reserved	Reserved	RO	-
[56:48]	exp_raid_#{index*4+3}_ldid	Specifies the LDID for Expanded RAID #{index*4+3}	RW	9'h0
[47]	exp_raid_#{index*4+2}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+2} is valid	RW	1'b0
[46]	exp_raid_#{index*4+2}_ldid_rnf	Specifies if Expanded RAID #{index*4+2} is RN-F	RW	1'b0
[45:41]	Reserved	Reserved	RO	-
[40:32]	exp_raid_#{index*4+2}_ldid	Specifies the LDID for Expanded RAID #{index*4+2}	RW	9'h0
[31]	exp_raid_#{index*4+1}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4+1} is valid	RW	1'b0
[30]	exp_raid_#{index*4+1}_ldid_rnf	Specifies if Expanded RAID #{index*4+1} is RN-F	RW	1'b0
[29:25]	Reserved	Reserved	RO	-
[24:16]	exp_raid_#{index*4+1}_ldid	Specifies the LDID for Expanded RAID #{index*4+1}	RW	9'h0
[15]	exp_raid_#{index*4}_ldid_valid	Specifies if LDID for Expanded RAID #{index*4} is valid	RW	1'b0
[14]	exp_raid_#{index*4}_ldid_rnf	Specifies if Expanded RAID #{index*4} is RN-F	RW	1'b0
[13:9]	Reserved	Reserved	RO	-
[8:0]	exp_raid_#{index*4}_ldid	Specifies the LDID for Expanded RAID #{index*4}	RW	9'h0

4.3.2.23 por_ccg_ha_pmu_event_sel

Specifies the PMU event to be counted as a 8-bit ID with the following encodings:

8'h00 CXHA_PMU_EVENT_NULL
8'h61 CXHA_PMU_EVENT_RDDATBYP
8'h62 CXHA_PMU_EVENT_CHIRSP_UP_STALL
8'h63 CXHA_PMU_EVENT_CHIDAT_UP_STALL
8'h64 CXHA_PMU_EVENT_SNPPCRD_LNK0_STALL
8'h65 CXHA_PMU_EVENT_SNPPCRD_LNK1_STALL
8'h66 CXHA_PMU_EVENT_SNPPCRD_LNK2_STALL
8'h67 CXHA_PMU_EVENT_REQTRK_OCC
8'h68 CXHA_PMU_EVENT_RDB_OCC
8'h69 CXHA_PMU_EVENT_RDBBYP_OCC
8'h6A CXHA_PMU_EVENT_WDB_OCC
8'h6B CXHA_PMU_EVENT_SNPTRK_OCC
8'h6C CXHA_PMU_EVENT_SDB_OCC
8'h6D CXHA_PMU_EVENT_SNPHAZ_OCC
8'h6E CXHA_PMU_EVENT_REQTRK_ALLOC
8'h6F CXHA_PMU_EVENT_RDB_ALLOC
8'h70 CXHA_PMU_EVENT_RDBBYP_ALLOC
8'h71 CXHA_PMU_EVENT_WDB_ALLOC
8'h72 CXHA_PMU_EVENT_SNPTRK_ALLOC
8'h73 CXHA_PMU_EVENT_SDB_ALLOC
8'h74 CXHA_PMU_EVENT_SNPHAZ_ALLOC

8'h75	CCHA_PMU_EVENT_PB_RHU_REQ_OCC
8'h76	CCHA_PMU_EVENT_PB_RHU_REQ_ALLOC
8'h77	CCHA_PMU_EVENT_PB_RHU_PCIE_REQ_OCC
8'h78	CCHA_PMU_EVENT_PB_RHU_PCIE_REQ_ALLOC
8'h79	CCHA_PMU_EVENT_PB_PCIE_WR_REQ_OCC
8'h7A	CCHA_PMU_EVENT_PB_PCIE_WR_REQ_ALLOC
8'h7B	CCHA_PMU_EVENT_PB_PCIE_REG_REQ_OCC
8'h7C	CCHA_PMU_EVENT_PB_PCIE_REG_REQ_ALLOC
8'h7D	CCHA_PMU_EVENT_PB_PCIE_RSVD_REQ_OCC
8'h7E	CCHA_PMU_EVENT_PB_PCIE_RSVD_REQ_ALLOC
8'h7F	CCHA_PMU_EVENT_PB_RHU_DAT_OCC
8'h80	CCHA_PMU_EVENT_PB_RHU_DAT_ALLOC
8'h81	CCHA_PMU_EVENT_PB_RHU_PCIE_DAT_OCC
8'h82	CCHA_PMU_EVENT_PB_RHU_PCIE_DAT_ALLOC
8'h83	CCHA_PMU_EVENT_PB_PCIE_WR_DAT_OCC
8'h84	CCHA_PMU_EVENT_PB_PCIE_WR_DAT_ALLOC

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-29: por_ccg_ha_pmu_event_sel

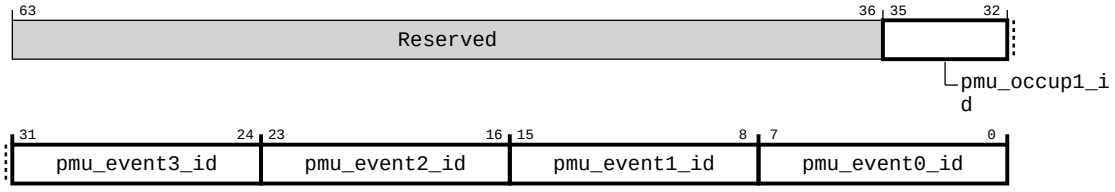


Table 4-45: por_ccg_ha_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	-
[35:32]	pmu_occup1_id	CXHA PMU occupancy event selector ID	RW	4'b0
[31:24]	pmu_event3_id	CXHA PMU Event 3 ID	RW	8'b0
[23:16]	pmu_event2_id	CXHA PMU Event 2 ID	RW	8'b0
[15:8]	pmu_event1_id	CXHA PMU Event 1 ID	RW	8'b0
[7:0]	pmu_event0_id	CXHA PMU Event 0 ID	RW	8'b0

4.3.2.24 por_ccg_ha_cxprtcl_link0_ctl

Functions as the CXHA CCIX Protocol Link 0 control register. Works with por_ccg_ha_cxprtcl_link0_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C00`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-30: por_ccg_ha_cxprtcl_link0_ctl

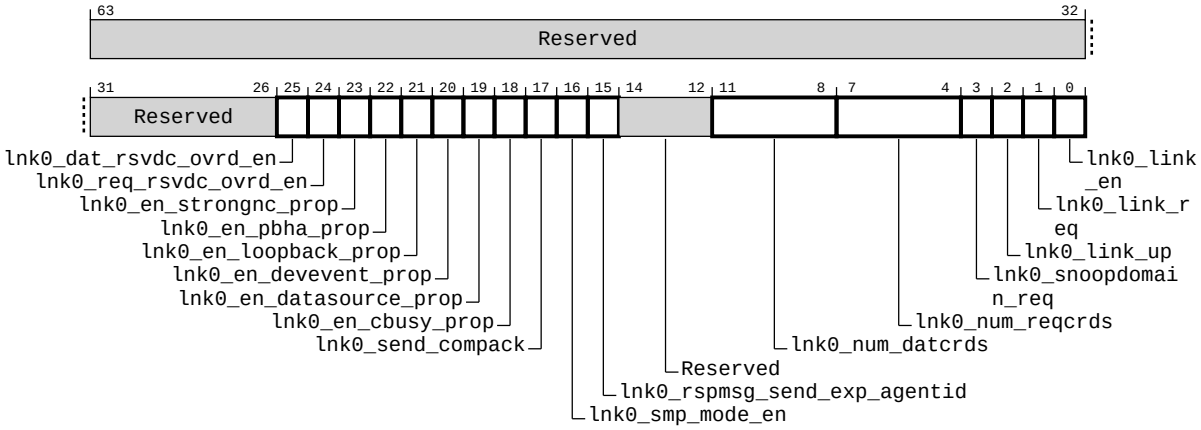


Table 4-46: por_ccg_ha_cxprtcl_link0_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	lnk0_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 0. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[24]	lnk0_req_rsvdc_ovrd_en	When set, overrides CHI REQ RSVDC field with req rsvdc strap input for incoming requests on CCIX Link 0. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[23]	lnk0_en_strongnc_prop	When set, enables propagation of StrongNC on CCIX Link 0.	RW	1'b0
[22]	lnk0_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 0.	RW	1'b0
[21]	lnk0_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 0.	RW	1'b0
[20]	lnk0_en_devevent_prop	When set, enables propagation of DevEvent on CCIX Link 0.	RW	1'b1
[19]	lnk0_en_datasource_prop	When set, enables propagation of DataSource on CCIX Link 0.	RW	1'b1
[18]	lnk0_en_cbusy_prop	When set, enables propagation of CBusy on CCIX Link 0.	RW	1'b1
[17]	lnk0_send_compack	When set, sends CompAck for CCIX Link 0.	RW	1'b0
[16]	lnk0_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 0.	RW	Configuration dependent
[15]	lnk0_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 0	RW	1'b0
[14:12]	Reserved	Reserved	RO	-
[11:8]	lnk0_num_datcrds	Controls the number of CCIX data credits assigned to Link 0 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0

Bits	Name	Description	Type	Reset
[7:4]	lnk0_num_reqcrds	Controls the number of CCIX request credits assigned to Link 0 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[3]	lnk0_snoopdomain_req	Controls Snoop domain enable (SYSCOREQ) for CCIX Link 0	RW	1'b0
[2]	lnk0_link_up	Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent **1'b0** Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear **1'b1** Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent	RW	1'b0
[1]	lnk0_link_req	Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent **1'b0** Link Down request **1'b1** Link Up request The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.	RW	1'b0
[0]	lnk0_link_en	Enables CCIX Link 0 when set **1'b0** Link is disabled **1'b1** Link is enabled	RW	1'b0

4.3.2.25 por_ccg_ha_cxprtcl_link0_status

Functions as the CXHA CCIX Protocol Link 0 status register. Works with por_ccg_ha_cxprtcl_link0_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C08`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-31: por_ccg_ha_cxprtcl_link0_status

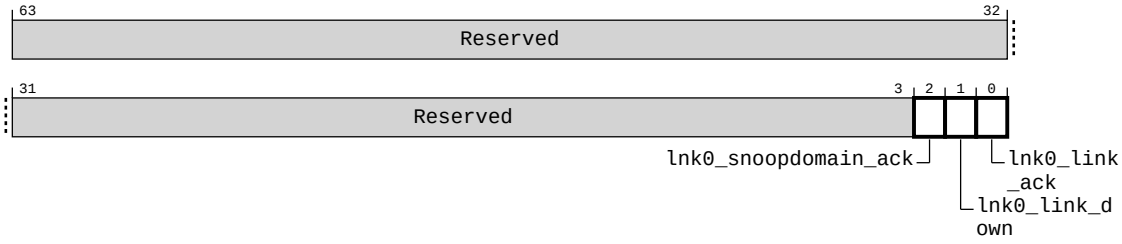


Table 4-47: por_ccg_ha_cxprtcl_link0_status attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	lnk0_snoopdomain_ack	Provides Snoop domain status (SYSCOACK) for CCIX Link 0	RO	1'b0
[1]	lnk0_link_down	Link Down status; hardware updates this register bit to indicate Link Down status **1'b0** Link is not Down; hardware clears Link_DN when it receives a Link Up request **1'b1** Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	1'b1
[0]	lnk0_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request **1'b0** Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear **1'b1** Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent NOTE: The local agent must clear Link_DN before setting Link_ACK.	RO	1'b0

4.3.2.26 por_ccg_ha_cxprtcl_link1_ctl

Functions as the CXHA CCIX Protocol Link 1 control register. Works with por_ccg_ha_cxprtcl_link1_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C10`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-32: por_ccg_ha_cxprtcl_link1_ctl

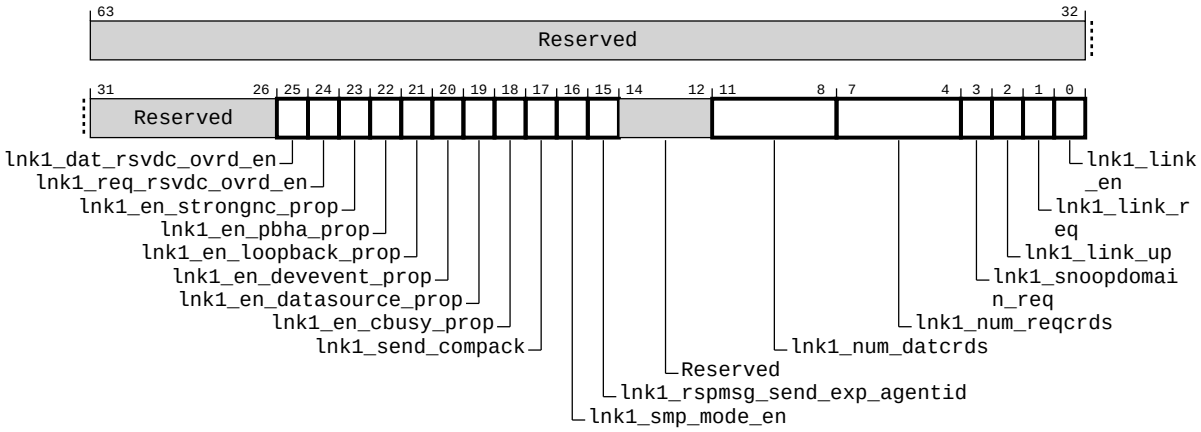


Table 4-48: por_ccg_ha_cxprtcl_link1_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[25]	lnk1_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 1. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[24]	lnk1_req_rsvdc_ovrd_en	When set, overrides CHI REQ RSVDC field with req rsvdc strap input for incoming requests on CCIX Link 1. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[23]	lnk1_en_strongnc_prop	When set, enables propagation of StrongNC on CCIX Link 1.	RW	1'b0
[22]	lnk1_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 1.	RW	1'b0
[21]	lnk1_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 1.	RW	1'b0
[20]	lnk1_en_devevent_prop	When set, enables propagation of DevEvent on CCIX Link 1.	RW	1'b1
[19]	lnk1_en_datasource_prop	When set, enables propagation of DataSource on CCIX Link 1.	RW	1'b1
[18]	lnk1_en_cbusy_prop	When set, enables propagation of CBusy on CCIX Link 1.	RW	1'b1
[17]	lnk1_send_compack	When set, sends CompAck for CCIX Link 1.	RW	1'b0
[16]	lnk1_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 1.	RW	Configuration dependent
[15]	lnk1_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 1	RW	1'b0
[14:12]	Reserved	Reserved	RO	-
[11:8]	lnk1_num_datcrds	Controls the number of CCIX data credits assigned to Link 1 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[7:4]	lnk1_num_reqcrds	Controls the number of CCIX request credits assigned to Link 1 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[3]	lnk1_snoopdomain_req	Controls Snoop domain enable (SYSCOREQ) for CCIX Link 1	RW	1'b0
[2]	lnk1_link_up	Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent **1'b0** Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear **1'b1** Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	lnk1_link_req	Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent **1'b0** Link Down request **1'b1** Link Up request The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.	RW	1'b0
[0]	lnk1_link_en	Enables CCIX Link 1 when set **1'b0** Link is disabled **1'b1** Link is enabled	RW	1'b0

4.3.2.27 por_ccg_ha_cxprtcl_link1_status

Functions as the CXHA CCIX Protocol Link 1 status register. Works with por_ccg_ha_cxprtcl_link1_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C18`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-33: por_ccg_ha_cxprtcl_link1_status

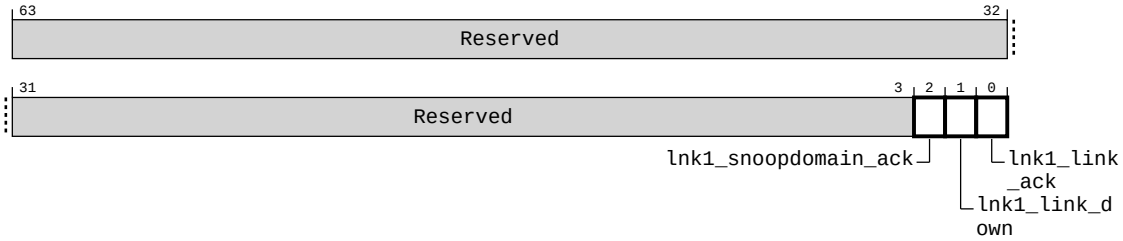


Table 4-49: por_ccg_ha_cxprtcl_link1_status attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	lnk1_snoopdomain_ack	Provides Snoop domain status (SYSCOACK) for CCIX Link 1	RO	1'b0
[1]	lnk1_link_down	Link Down status; hardware updates this register bit to indicate Link Down status **1'b0** Link is not Down; hardware clears Link_DN when it receives a Link Up request **1'b1** Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	1'b1
[0]	lnk1_link_ack	Link Up/Down Acknowledge; hardware updates this register bit to acknowledge the software link request **1'b0** Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear **1'b1** Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent NOTE: The local agent must clear Link_DN before setting Link_ACK.	RO	1'b0

4.3.2.28 por_ccg_ha_cxprtcl_link2_ctl

Functions as the CXHA CCIX Protocol Link 2 control register. Works with por_ccg_ha_cxprtcl_link2_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C20`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-34: por_ccg_ha_cxprtcl_link2_ctl

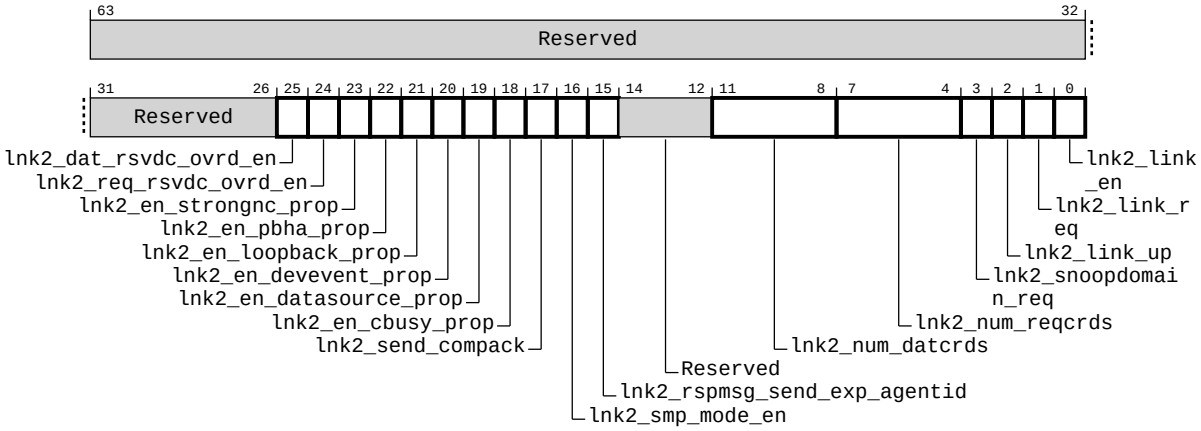


Table 4-50: por_ccg_ha_cxprtcl_link2_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	lnk2_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 2. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[24]	lnk2_req_rsvdc_ovrd_en	When set, overrides CHI REQ RSVDC field with req rsvdc strap input for incoming requests on CCIX Link 2. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[23]	lnk2_en_strongnc_prop	When set, enables propagation of StrongNC on CCIX Link 2.	RW	1'b0
[22]	lnk2_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 2.	RW	1'b0
[21]	lnk2_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 2.	RW	1'b0
[20]	lnk2_en_devevent_prop	When set, enables propagation of DevEvent on CCIX Link 2.	RW	1'b1
[19]	lnk2_en_datasource_prop	When set, enables propagation of DataSource on CCIX Link 2.	RW	1'b1
[18]	lnk2_en_cbusy_prop	When set, enables propagation of CBusy on CCIX Link 2.	RW	1'b1
[17]	lnk2_send_compack	When set, sends CompAck for CCIX Link 2.	RW	1'b0
[16]	lnk2_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 2.	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[15]	lnk2_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 2	RW	1'b0
[14:12]	Reserved	Reserved	RO	-
[11:8]	lnk2_num_datcrds	Controls the number of CCIX data credits assigned to Link 2 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[7:4]	lnk2_num_reqcrds	Controls the number of CCIX request credits assigned to Link 2 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[3]	lnk2_snoopdomain_req	Controls Snoop domain enable (SYSCOREQ) for CCIX Link 2	RW	1'b0
[2]	lnk2_link_up	Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent **1'b0** Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear **1'b1** Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent	RW	1'b0
[1]	lnk2_link_req	Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent **1'b0** Link Down request **1'b1** Link Up request The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.	RW	1'b0
[0]	lnk2_link_en	Enables CCIX Link 2 when set **1'b0** Link is disabled **1'b1** Link is enabled	RW	1'b0

4.3.2.29 por_ccg_ha_cxprtcl_link2_status

Functions as the CXHA CCIX Protocol Link 2 status register. Works with por_ccg_ha_cxprtcl_link2_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C28`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-35: por_ccg_ha_cxprtcl_link2_status

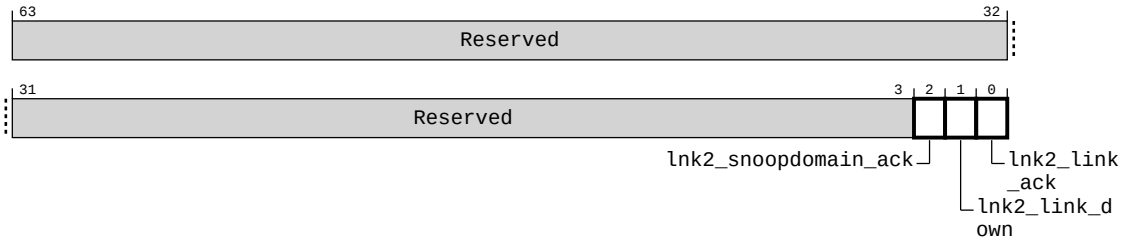


Table 4-51: por_ccg_ha_cxprtcl_link2_status attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	lnk2_snoopdomain_ack	Provides Snoop domain status (SYSCOACK) for CCIX Link 2	RO	1'b0
[1]	lnk2_link_down	Link Down status; hardware updates this register bit to indicate Link Down status **1'b0** Link is not Down; hardware clears Link_DN when it receives a Link Up request **1'b1** Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	1'b1

Bits	Name	Description	Type	Reset
[0]	lnk2_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request **1'b0** Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear **1'b1** Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent NOTE: The local agent must clear Link_DN before setting Link_ACK.	RO	1'b0

4.3.3 CCG_RA register descriptions

This section lists the CCG_RA registers.

4.3.3.1 por_ccg_ra_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-36: por_ccg_ra_node_info

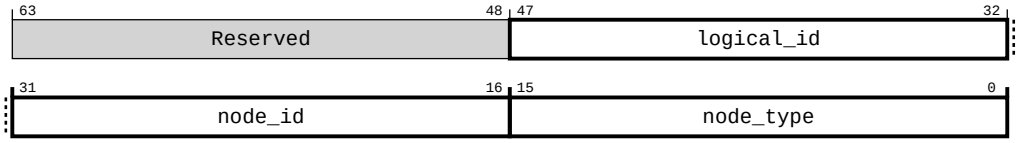


Table 4-52: por_ccg_ra_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0103

4.3.3.2 por_ccg_ra_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-37: por_ccg_ra_child_info

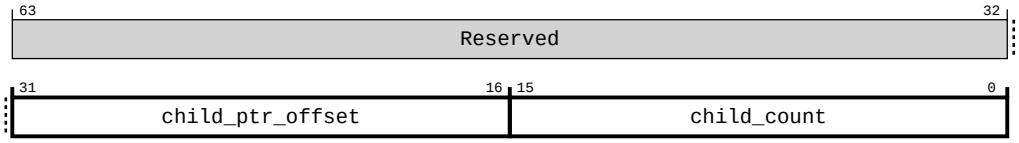


Table 4-53: por_ccg_ra_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'h0

4.3.3.3 por_ccg_ra_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-38: por_ccg_ra_secure_register_groups_override

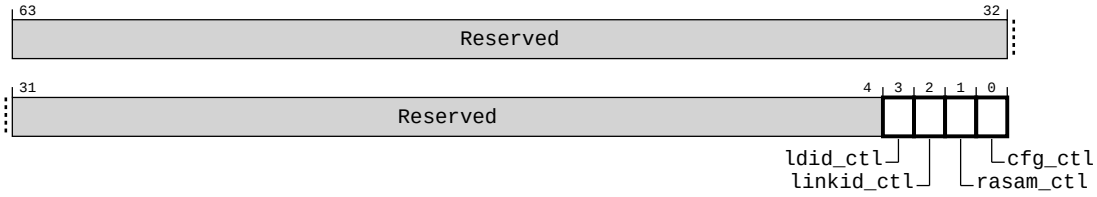


Table 4-54: por_ccg_ra_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	ldid_ctl	Allows Non-secure access to Secure RA LDID registers	RW	1'b0
[2]	linkid_ctl	Allows Non-secure access to Secure RA Link ID registers	RW	1'b0
[1]	rasam_ctl	Allows Non-secure access to Secure RA SAM control registers	RW	1'b0
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register	RW	1'b0

4.3.3.4 por_ccg_ra_unit_info

Provides component identification information for CXRA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-39: por_ccg_ra_unit_info

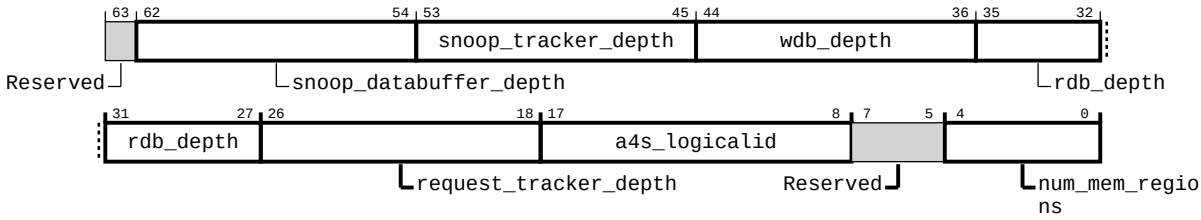


Table 4-55: por_ccg_ra_unit_info attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:54]	snoop_databuffer_depth	Depth of Snoop Data Buffer - number of outstanding SNP requests on CHI	RO	Configuration dependent
[53:45]	snoop_tracker_depth	Depth of Snoop Tracker - number of outstanding SNP requests on CCIX	RO	Configuration dependent
[44:36]	wdb_depth	Depth of Write Data Buffer	RO	Configuration dependent
[35:27]	rdb_depth	Depth of Read Data Buffer	RO	Configuration dependent
[26:18]	request_tracker_depth	Depth of Request Tracker - number of outstanding Memory requests on CCIX	RO	Configuration dependent
[17:8]	a4s_logicalid	AXI4Stream interfaces logical ID	RO	Configuration dependent
[7:5]	Reserved	Reserved	RO	-
[4:0]	num_mem_regions	Number of memory regions supported	RO	Configuration dependent

4.3.3.5 por_ccg_ra_cfg_ctl

Functions as the configuration control register. Specifies the current mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-40: por_ccg_ra_cfg_ctl

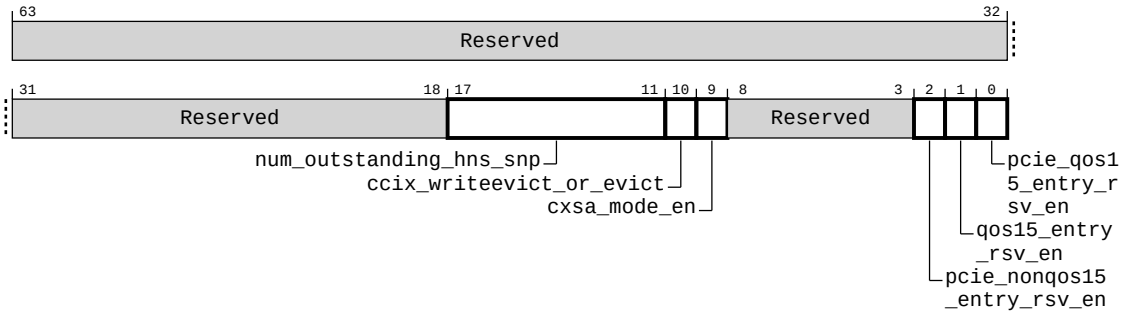


Table 4-56: por_ccg_ra_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:18]	Reserved	Reserved	RO	-
[17:11]	num_outstanding_hns_snp	Specifies the Max number of outstanding snoops from the given RA to each HNS to guarantee snoop sink and deadlock prevention. Must be set to (HNS_NUM_ENTRIES_SNPQ_PARAM)/(NUM_NON_CXSA_RA).	RW	7'h2
[10]	ccix_writeevict_or_evict	When set, downgrades WriteEvictOrEvict to Evict **1'b1** Evict is sent **1'b0** WriteEvict is sent	RW	1'b0
[9]	cxsa_mode_en	When set, enables the CCIX Subordinate Agent mode.In this mode RA functions as a CCIX Subordinate Agent **1'b1** CCIX Subordinate Agent **1'b0** CCIX Requesting Agent	RW	1'b0
[8:3]	Reserved	Reserved	RO	-
[2]	pcie_nonqos15_entry_rsv_en	Enables entry reservation for non QoS15 traffic from PCIe RN-I/RN-D **1'b1** Reserves tracker entry for non QoS15 requests from PCIe RN-I/RN-D **1'b0** Does not reserve tracker entry for non QoS15 requests from PCIe RN-I/RN-D	RW	1'b1

Bits	Name	Description	Type	Reset
[1]	qos15_entry_rsv_en	Enables entry reservation for QoS15 traffic **1'b1** Reserves tracker entry for QoS15 requests **1'b0** Does not reserve tracker entry for QoS15 requests	RW	1'b1
[0]	pcie_qos15_entry_rsv_en	Enables entry reservation for QoS15 traffic from PCIe RN-I/RN-D **1'b1** Reserves tracker entry for QoS15 requests from PCIe RN-I/RN-D **1'b0** Does not reserve tracker entry for QoS15 requests from PCIe RN-I/RN-D	RW	1'b1

4.3.3.6 por_ccg_ra_aux_ctl

Functions as the auxiliary control register for CXRA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-41: por_ccg_ra_aux_ctl

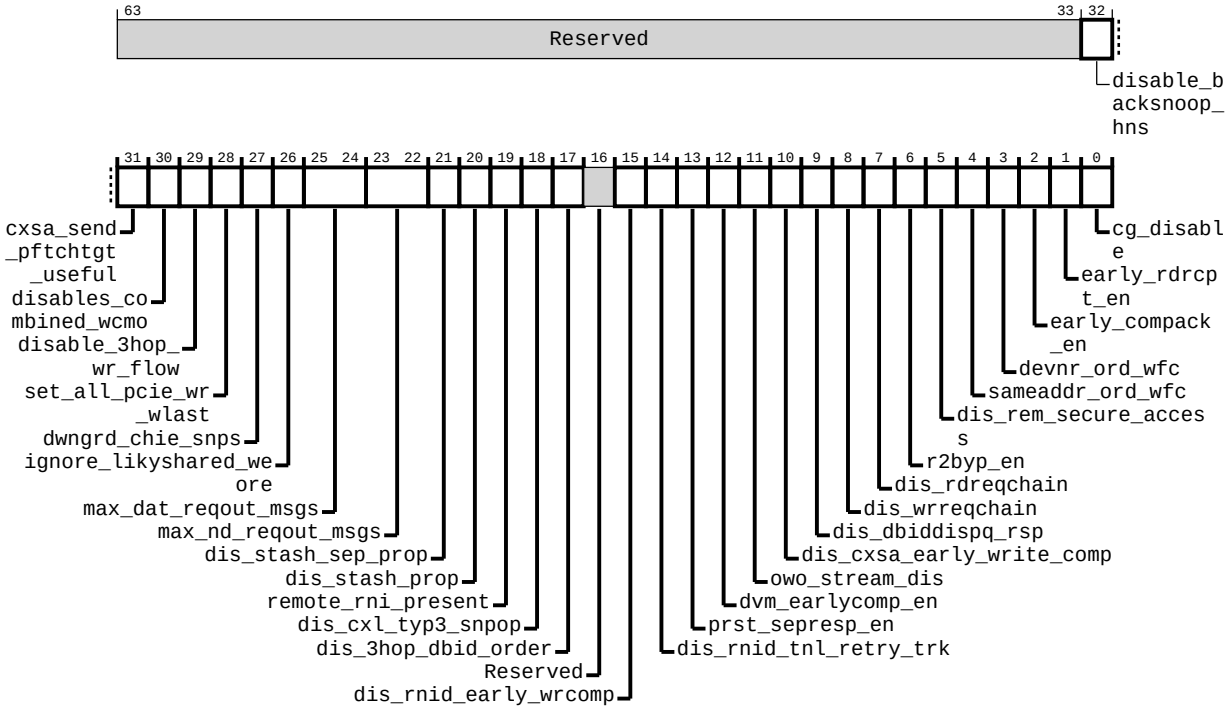


Table 4-57: por_ccg_ra_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:33]	Reserved	Reserved	RO	-
[32]	disable_backsnoop_hns	When set, disables back snooping on an HNS/LCN initiated WriteClean request which is due to LLC eviction	RW	1'b1
[31]	cxsa_send_pftchtgt_useful	When set, sends PreFetch Usefull indication on CHI datasource field in CXSA mode	RW	1'b0
[30]	disables_combined_wcmo	When set, disables sending combined W+CMO to remote HA. Applicable only in SMP mode	RW	1'b1
[29]	disable_3hop_wr_flow	When set, disables 3-hop write flow to remote HA. Applicable only in SMP mode	RW	1'b0
[28]	set_all_pcie_wr_wlast	When set, sets WLAST indication for all PCIe writes going to HA RNI	RW	1'b0
[27]	dwngrd_chie_snps	When set, downgrades CHIE SnpPreferUnique to SnpNotSharedDirty	RW	1'b0
[26]	ignore_likyshared_weore	When set, disables the use of LikelyShared(LS) bit to make a decision for WriteEvictOrEvict **1'b0** Send WriteEvict when LS= 0 and send Evict when LS=1 **1'b1** Ignore LS bit. WriteEvict is sent. Further static decision can be made using ccix_writeevict_or_evict in cfg_ctl register	RW	1'b0

Bits	Name	Description	Type	Reset
[25:24]	max_dat_reqout_msgs	Used to configure the maximum number of data requests messages (writes, atomics etc.) presented to CCLA's packing logic. **2'b00** one message **2'b01** two messages **2'b10** three messages **2'b11** four messages Note: The max is further limited by max allowed by the given protocol (CCIX2.0/ CXL.mem/CXL.cache	RW	2'b11
[23:22]	max_nd_reqout_msgs	Used to configure the maximum number of non-data requests messages (reads, dataless) presented to CCLA's packing logic. **2'b00** one message **2'b01** two messages **2'b10** three messages **2'b11** four messages Note: The max is further limited by max allowed by the given protocol (CCIX2.0/ CXL.mem/CXL.cache	RW	2'b11
[21]	dis_stash_sep_prop	When set, disables propagation of StashSep opcodes on CCIX. StashSep opcodes are sent as Stash opcodes when set. Applicable only in SMP mode	RW	1'b0
[20]	dis_stash_prop	When set, disables propagation of stash opcodes on CCIX. Applicable only in SMP mode	RW	1'b0
[19]	remote_rni_present	When set, Enables TXNID coloring to enable traffic to remote RNI	RW	1'b1
[18]	dis_cxl_typ3_snpop	When set, drives SnpType= NOP on CXL Type3 M2S Req and RwD messages	RW	1'b1
[17]	dis_3hop_dbid_order	When set, disables ordered disapatch of DBIDs for 3-hop writes. By default, 3-hop DBIDs are dispatched in order. Applicable only if 3-hop write flow is enabled	RW	1'b0
[16]	Reserved	Reserved	RO	-
[15]	dis_rnid_early_wrcomp	When set, disables early write completions for tunneled writes from RNI.	RW	1'b0
[14]	dis_rnid_tnl_retry_trk	When set, disables RNID write request tunneling retry tracker.	RW	1'b0
[13]	prst_sepresp_en	When set, enables separate persist response on CCIX for persistent cache maintenance (PCMO2) operation Note: this bit is applicable only in SMP mode.	RW	1'b1
[12]	dvm_earlycomp_en	When set, enables early DVM Op completion responses from RA.	RW	1'b1
[11]	owo_stream_dis	When set, disables CompAck dependency to dispatch an ordered PCIe write.	RW	1'b1
[10]	dis_cxsa_early_write_comp	When set, disables early write completions in CCIX Subordinate Agent mode.	RW	1'b0
[9]	dis_dbiddispq_rsp	When set, disables the dispatch of DBID responses from a separate DispatchQ.	RW	1'b0
[8]	dis_wrreqchain	When set, disables chaining of write requests.	RW	1'b0
[7]	dis_rdreqchain	When set, disables chaining of read and dataless requests.	RW	1'b0
[6]	r2byp_en	When set, enables request bypass. Applies to read and dataless requests only. Note: When set will affect the capability to chain a request on the TX side	RW	1'b1
[5]	dis_rem_secure_access	When set, treats all the incoming snoops as Non-secure and forces the NS bit to 1	RW	1'b0
[4]	sameaddr_ord_wfc	When set, enables waiting for completion (COMP) before dispatching next same Addr dependent transaction (TXN)	RW	1'b0
[3]	devnr_ord_wfc	When set, enables waiting for completion (COMP) before dispatching next Device-nR dependent transaction (TXN)	RW	1'b0
[2]	early_compack_en	Early CompAck enable; enables sending early CompAck on CCIX for requests that require CompAck	RW	1'b1

Bits	Name	Description	Type	Reset
[1]	early_rdrcpt_en	Early ReadReceipt enable; enables sending early ReadReceipt for ordered read requests	RW	1'b1
[0]	cg_disable	Disables clock gating when set	RW	1'b0

4.3.3.7 por_ccg_ra_cbusy_limit_ctl

Cbusy threshold limits for RHT entries.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA18`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-42: por_ccg_ra_cbusy_limit_ctl

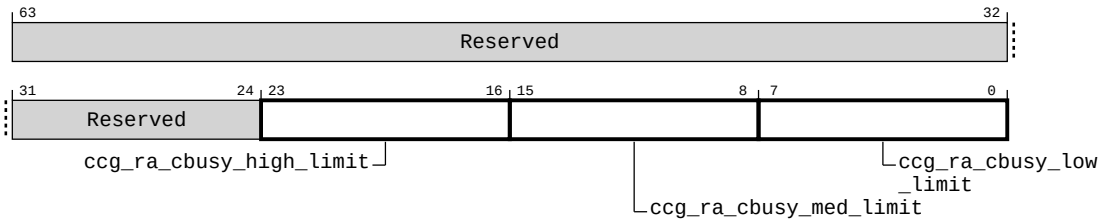


Table 4-58: por_ccg_ra_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	ccg_ra_cbusy_high_limit	RHT limit for CBusy High	RW	Configuration dependent
[15:8]	ccg_ra_cbusy_med_limit	RHT limit for CBusy Med	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[7:0]	ccg_ra_cbusy_low_limit	RHT limit for CBusy Low	RW	Configuration dependent

4.3.3.8 por_ccg_ra_sam_addr_reg0-7on_reg0-7ndex

There are 8 iterations of this register. The index ranges from 0 to 7. Configures Address Region #{index} for RA SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.rasam_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-43: por_ccg_ra_sam_addr_reg0-7on_reg0-7ndex

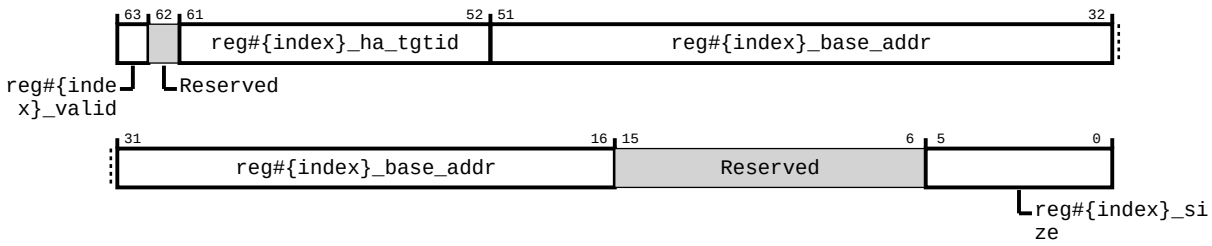


Table 4-59: por_ccg_ra_sam_addr_reg0-7on_reg0-7ndex attributes

Bits	Name	Description	Type	Reset
[63]	reg#{index}_valid	Specifies if the memory region is valid	RW	1'b0

Bits	Name	Description	Type	Reset
[62]	Reserved	Reserved	RO	-
[61:52]	reg#{index}_ha_tgtid	Specifies the target HAID	RW	10'b0
[51:16]	reg#{index}_base_addr	Specifies the 2^n-aligned base address for the memory region	RW	36'h0
[15:6]	Reserved	Reserved	RO	-
[5:0]	reg#{index}_size	Specifies the size of the memory region	RW	1'b0

4.3.3.9 por_ccg_ra_agentid_to_linkid_val

Specifies which Agent ID to Link ID mappings are valid.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-44: por_ccg_ra_agentid_to_linkid_val

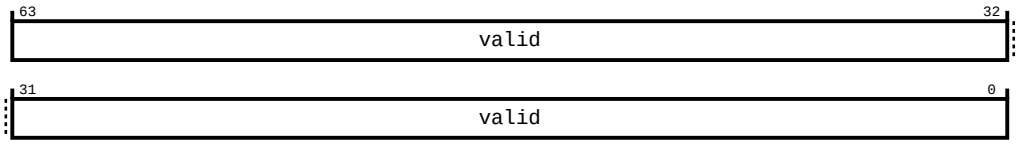


Table 4-60: por_ccg_ra_agentid_to_linkid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Link ID is valid; bit number corresponds to logical Agent ID number (from 0 to 63)	RW	63'h0

4.3.3.10 por_ccg_ra_agent0-7d_to_l0-7nk0-7d_reg0-7ndex

There are 8 iterations of this register. The index ranges from 0 to 7. Specifies the mapping of Agent ID to Link ID for Agent IDs #{index*8} to #{index*8+7}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD10 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-45: por_ccg_ra_agent0-7d_to_l0-7nk0-7d_reg0-7ndex

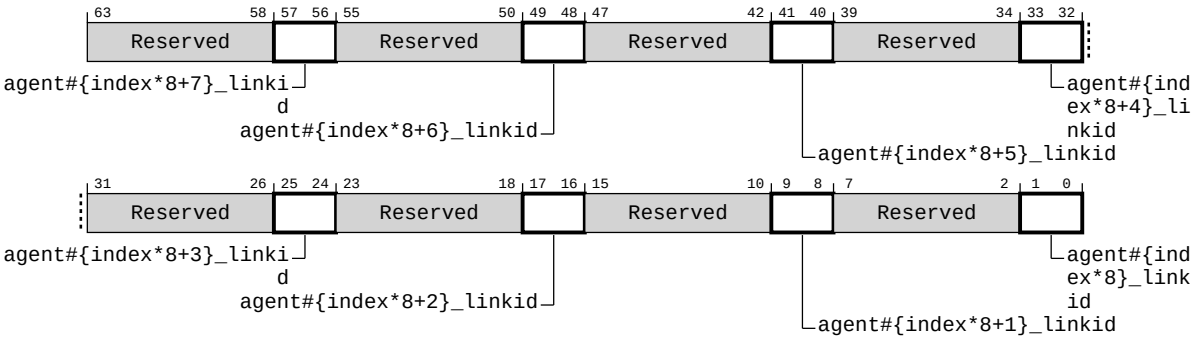


Table 4-61: por_ccg_ra_agent0-7d_to_l0-7nk0-7d_reg0-7ndex attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent#{index*8+7}_linkid	Specifies the Link ID for Agent ID #{index*8+7}	RW	2'h0

Bits	Name	Description	Type	Reset
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent#{index*8+6}_linkid	Specifies the Link ID for Agent ID #{index*8+6}	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent#{index*8+5}_linkid	Specifies the Link ID for Agent ID #{index*8+5}	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent#{index*8+4}_linkid	Specifies the Link ID for Agent ID #{index*8+4}	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent#{index*8+3}_linkid	Specifies the Link ID for Agent ID #{index*8+3}	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent#{index*8+2}_linkid	Specifies the Link ID for Agent ID #{index*8+2}	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent#{index*8+1}_linkid	Specifies the Link ID for Agent ID #{index*8+1}	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent#{index*8}_linkid	Specifies the Link ID for Agent ID #{index*8}	RW	2'h0

4.3.3.11 por_ccg_ra_rn0-31_ld0-31d_to_exp_ra0-31d_reg0-31ndex

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of RN-I's LDID to Expanded RAID for LDIDs #{i*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE00 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-46: por_ccg_ra_rn0-31_ld0-31d_to_exp_ra0-31d_reg0-31ndex

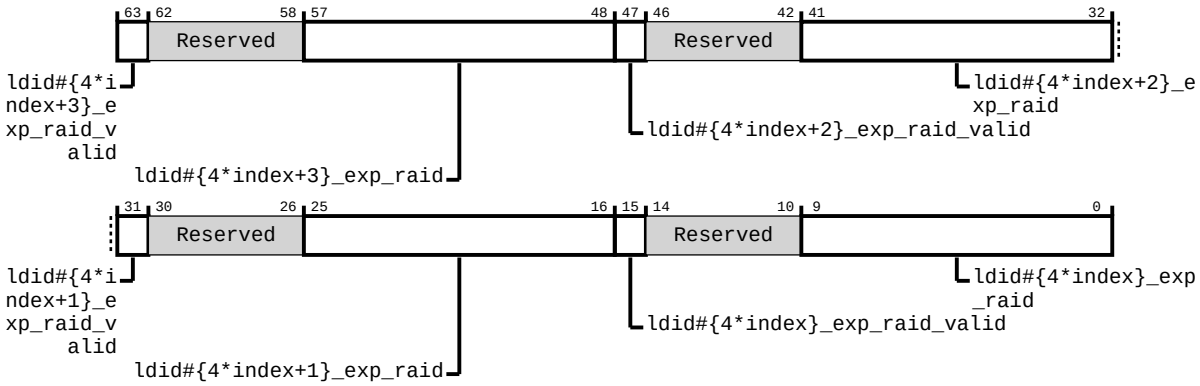


Table 4-62: por_ccg_ra_rn0-31_ld0-31d_to_exp_ra0-31d_reg0-31ndex attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{4*index+3}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-
[57:48]	ldid#{4*index+3}_exp_raid	Specifies the Expanded RAID for LDID #{4*index+3}	RW	10'h0
[47]	ldid#{4*index+2}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{4*index+2}_exp_raid	Specifies the Expanded RAID for LDID #{4*index+2}	RW	10'h0
[31]	ldid#{4*index+1}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{4*index+1}_exp_raid	Specifies the Expanded RAID for LDID #{4*index+1}	RW	10'h0
[15]	ldid#{4*index}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{4*index} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{4*index}_exp_raid	Specifies the Expanded RAID for LDID #{4*index}	RW	10'h0

4.3.3.12 por_ccg_ra_rnd_ld0-31d_to_exp_ra0-31d_reg0-31ndex

There are 32 iterations of this register. The index ranges from 0 to 31. Specifies the mapping of RN-D's LDID to Expanded RAID for LDIDs #{index*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'hF00 + #{index*8}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-47: por_ccg_ra_rnd_ld0-31d_to_exp_ra0-31d_reg0-31ndex

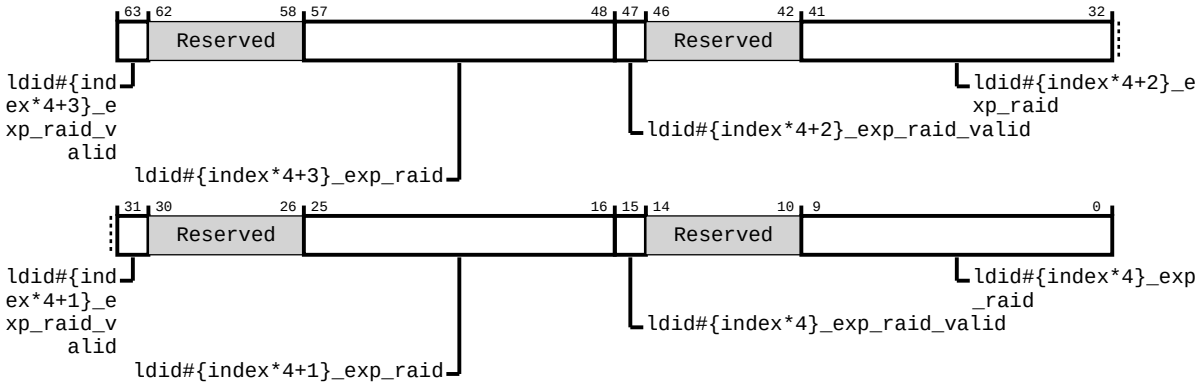


Table 4-63: por_ccg_ra_rnd_ld0-31d_to_exp_ra0-31d_reg0-31ndex attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{index*4+3}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{index*4+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-
[57:48]	ldid#{index*4+3}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+3}	RW	10'h0
[47]	ldid#{index*4+2}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{index*4+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{index*4+2}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+2}	RW	10'h0
[31]	ldid#{index*4+1}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{index*4+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{index*4+1}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+1}	RW	10'h0
[15]	ldid#{index*4}_exp_raid_valid	Specifies whether the Expanded RAID for LDID#{index*4} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{index*4}_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	10'h0

4.3.3.13 por_ccg_ra_rnf_ld0-127d_to_exp_ra0-127d_reg0-127ndex

There are 128 iterations of this register. The index ranges from 0 to 127. Specifies the mapping of RN-F's LDID to Expanded RAID for LDIDs #{index*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1000 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-48: por_ccg_ra_rnf_ld0-127d_to_exp_ra0-127d_reg0-127ndex

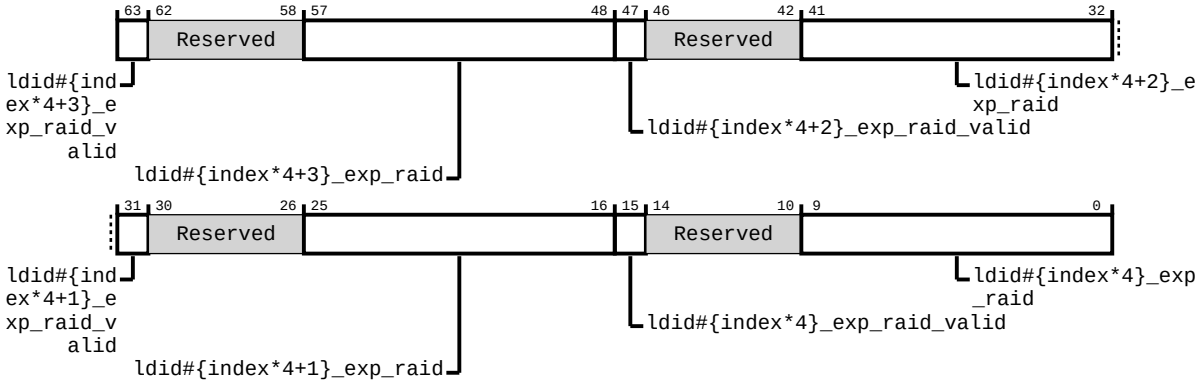


Table 4-64: por_ccg_ra_rnf_ld0-127d_to_exp_ra0-127d_reg0-127ndex attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{index*4+3}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[57:48]	ldid#{index*4+3}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+3}	RW	10'h0
[47]	ldid#{index*4+2}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{index*4+2}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+2}	RW	10'h0
[31]	ldid#{index*4+1}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{index*4+1}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+1}	RW	10'h0
[15]	ldid#{index*4}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{index*4}_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	10'h0

4.3.3.14 por_ccg_ra_ha_ld0-15d_to_exp_ra0-15d_reg0-15ndex

There are 16 iterations of this register. The index ranges from 0 to 15. Specifies the mapping of HA's LDID to Expanded RAID for LDIDs #{index*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1400 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-49: por_ccg_ra_ha_ld0-15d_to_exp_ra0-15d_reg0-15ndex

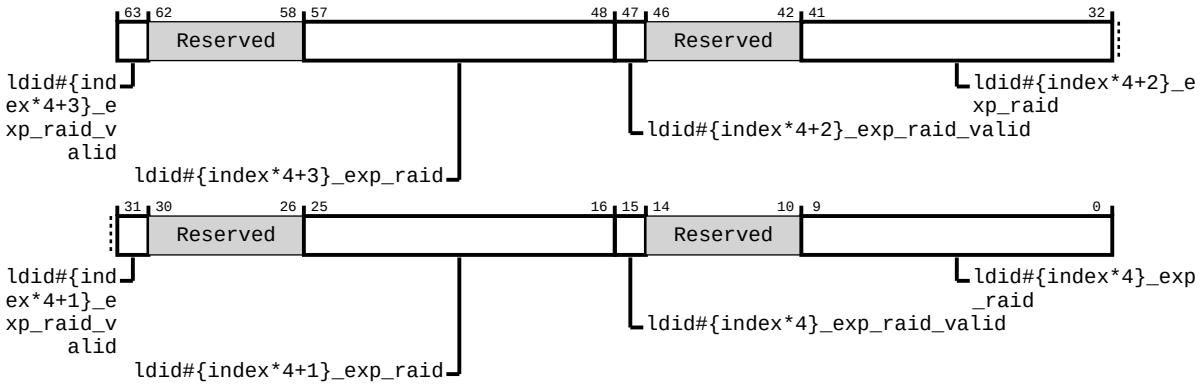


Table 4-65: por_ccg_ra_ha_ld0-15d_to_exp_ra0-15d_reg0-15ndex attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{index*4+3}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-
[57:48]	ldid#{index*4+3}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+3}	RW	10'h0
[47]	ldid#{index*4+2}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{index*4+2}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+2}	RW	10'h0
[31]	ldid#{index*4+1}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{index*4+1}_exp_raid	Specifies the Expanded RAID for LDID #{index*4+1}	RW	10'h0
[15]	ldid#{index*4}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{index*4}_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	10'h0

4.3.3.15 por_ccg_ra_hns_ld0-0d_to_exp_ra0-0d_reg0-0ndex

There are 1 iterations of this register. The index ranges from 0 to 0. Specifies the mapping of HNS's LDID to Expanded RAID for LDIDs #{index*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1480 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-50: por_ccg_ra_hns_ld0-0d_to_exp_ra0-0d_reg0-0ndex

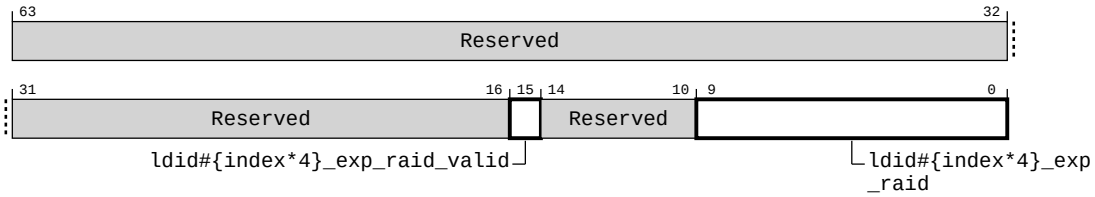


Table 4-66: por_ccg_ra_hns_ld0-0d_to_exp_ra0-0d_reg0-0ndex attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15]	ldid#{index*4}_exp_raid_valid	Specifies whether the look table entry for default LDID#{index*4} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{index*4}_exp_raid	Specifies the Expanded RAID for LDID #{index*4}	RW	10'h0

4.3.3.16 por_ccg_ra_rnf_ld0-127d_to_ovrd_ld0-127d_reg0-127ndex

There are 128 iterations of this register. The index ranges from 0 to 127. Specifies the mapping of RN-F's overridden LDID for default LDIDs #{index*4} to #{index*4+3}.Valid only if POR_MXP_RNF_CLUSTER_EN_PARAM is 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1500 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-51: por_ccg_ra_rnf_ld0-127d_to_ovrd_ld0-127d_reg0-127ndex

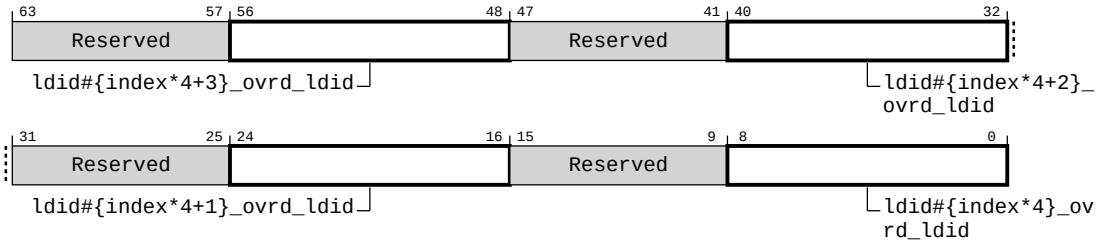


Table 4-67: por_ccg_ra_rnf_ld0-127d_to_ovrd_ld0-127d_reg0-127ndex attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ldid#{index*4+3}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4+3}	RW	Configuration dependent
[47:41]	Reserved	Reserved	RO	-
[40:32]	ldid#{index*4+2}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4+2}	RW	Configuration dependent
[31:25]	Reserved	Reserved	RO	-
[24:16]	ldid#{index*4+1}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4+1}	RW	Configuration dependent
[15:9]	Reserved	Reserved	RO	-
[8:0]	ldid#{index*4}_ovrd_ldid	Specifies the Overridden LDID for Default LDID #{index*4}	RW	Configuration dependent

4.3.3.17 por_ccg_ra_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-52: por_ccg_ra_pmu_event_sel

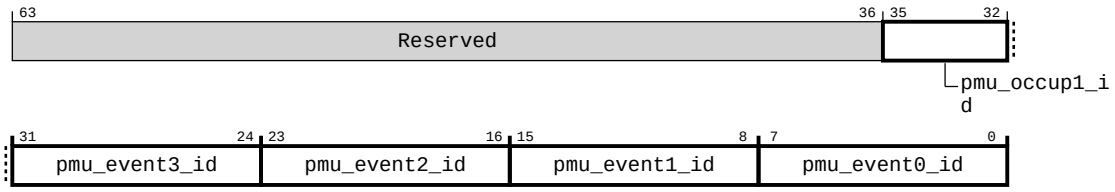


Table 4-68: por_ccg_ra_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	-
[35:32]	pmu_occup1_id	PMU occupancy event selector ID	RW	4'b0
[31:24]	pmu_event3_id	CXRA PMU Event 3 ID; see pmu_event0_id for encodings	RW	8'b0
[23:16]	pmu_event2_id	CXRA PMU Event 2 ID; see pmu_event0_id for encodings	RW	8'b0
[15:8]	pmu_event1_id	CXRA PMU Event 1 ID; see pmu_event0_id for encodings	RW	8'b0

Bits	Name	Description	Type	Reset
[7:0]	pmu_event0_id	CXRA PMU Event 0 ID **8'h00** No event **8'h41** Request Tracker (RHT) occupancy count overflow **8'h42** Snoop Tracker (SHT) occupancy count overflow **8'h43** Read Data Buffer (RDB) occupancy count overflow **8'h44** Write Data Buffer (WDB) occupancy count overflow **8'h45** Snoop Sink Buffer (SSB) occupancy count overflow **8'h46** CCIX RX broadcast snoops **8'h47** CCIX TX request chain **8'h48** CCIX TX request chain average length **8'h49** CHI internal RSP stall **8'h4A** CHI internal DAT stall **8'h4B** CCIX REQ Protocol credit Link 0 stall **8'h4C** CCIX REQ Protocol credit Link 1 stall **8'h4D** CCIX REQ Protocol credit Link 2 stall **8'h4E** CCIX DAT Protocol credit Link 0 stall **8'h4F** CCIX DAT Protocol credit Link 1 stall **8'h50** CCIX DAT Protocol credit Link 2 stall **8'h51** CHI external RSP stall **8'h52** CHI external DAT stall **8'h53** CCIX MISC Protocol credit Link 0 stall **8'h54** CCIX MISC Protocol credit Link 1 stall **8'h55** CCIX MISC Protocol credit Link 2 stall **8'h56** Request Tracker (RHT) allocations **8'h57** Snoop Tracker (SHT) allocations **8'h58** Read Data Buffer (RDB) allocations **8'h59** Write Data Buffer (WDB) allocations **8'h5A** Snoop Sink Buffer (SSB) allocations	RW	8'b0

4.3.3.18 por_ccg_ra_ccprtcl_link0_ctl

Functions as the CXRA CCIX Protocol Link 0 control register. Works with por_ccg_ra_ccprtcl_link0_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C00`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-53: por_ccg_ra_ccprtcl_link0_ctl

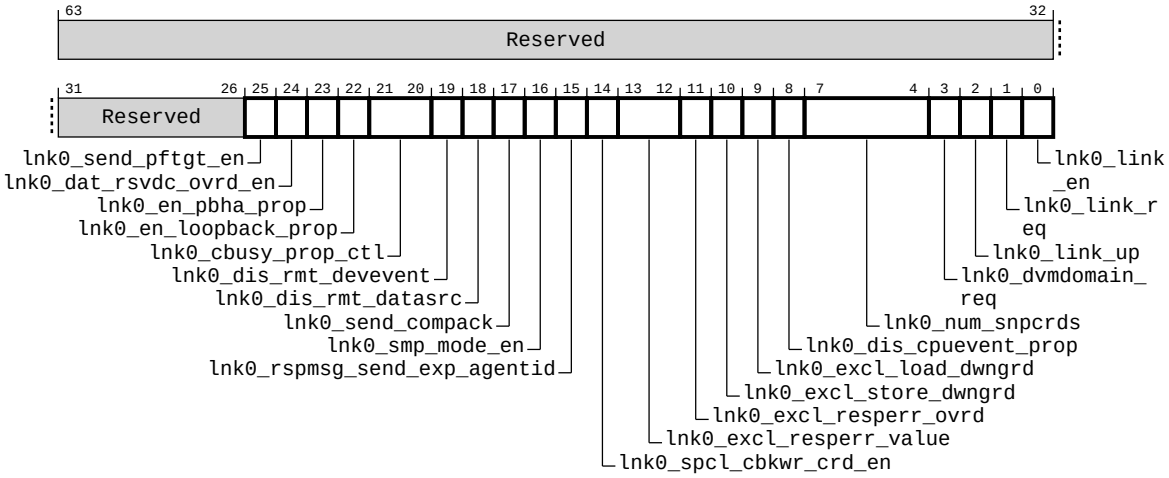


Table 4-69: por_ccg_ra_ccprtcl_link0_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	lnk0_send_pftgt_en	When set, enables sending Prefetch Target (CHI) or MemSpecRd (CXL) over link 0.	RW	1'b1
[24]	lnk0_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 0. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[23]	lnk0_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 0.	RW	1'b1
[22]	lnk0_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 0.	RW	1'b1
[21:20]	lnk0_cbusy_prop_ctl	Controls the propagation of Cbusy field for CCIX Link 0. **2'b00** Send RA Cbusy on all responses based on the limits programmed in por_ccg_ra_cbusy_limit_ctl **2'b01** Pass through remote CBusy on late completion responses (CompData, Comp) **2'b10** Greater of RA Cbusy or remote Cbusy. Applicable to responses where remote Cbusy can be sent NOTE: This field is applicable SMP and CXL modes	RW	2'b0
[19]	lnk0_dis_rmt_devevent	When set, disables propagation of remote Dev Event field for CCIX Link 0. NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0

Bits	Name	Description	Type	Reset
[18]	lnk0_dis_rmt_datasrc	When set, disables propagation of remote data source for CCIX Link 0. NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[17]	lnk0_send_compack	When set, sends CompAck for CCIX Link 0.	RW	1'b0
[16]	lnk0_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 0.	RW	Configuration dependent
[15]	lnk0_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 0	RW	1'b0
[14]	lnk0_spcl_cbkwr_crd_en	When set, notifies RA to use special credits from HA to send CopyBack writes on CCIX Link 0 NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[13:12]	lnk0_excl_resperr_value	Two bit value to override RespErr field of an exclusive response. Applicable only if lnk0_excl_resperr_ovrd bit is set. NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	2'b0
[11]	lnk0_excl_resperr_ovrd	When set, overrides the RespErr field of exclusive response with the lnk0_excl_resperr_value field NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[10]	lnk0_excl_store_dwngrd	When set, downgrades shareable exclusive store to shareable store when sending on CCIX Link 0 NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[9]	lnk0_excl_load_dwngrd	When set, downgrades shareable exclusive load to shareable load when sending on CCIX Link 0 NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[8]	lnk0_dis_cpuevent_prop	When set, disables the propagation of CPU Events on CCIX Link 0 NOTE: This field is applicable only when SMP Mode enable is set.	RW	1'b0
[7:4]	lnk0_num_snpcrds	Controls the number of CCIX snoop credits assigned to Link 0 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[3]	lnk0_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for CCIX Link 0	RW	1'b0
[2]	lnk0_link_up	Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent **1'b0** Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear **1'b1** Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	lnk0_link_req	Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent **1'b0** Link Down request **1'b1** Link Up request The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.	RW	1'b0
[0]	lnk0_link_en	Enables CCIX Link 0 when set **1'b0** Link is disabled **1'b1** Link is enabled	RW	1'b0

4.3.3.19 por_ccg_ra_ccprtcl_link0_status

Functions as the CXRA CCIX Protocol Link 0 status register. Works with por_ccg_ra_ccprtcl_link0_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C08`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-54: por_ccg_ra_ccprtcl_link0_status

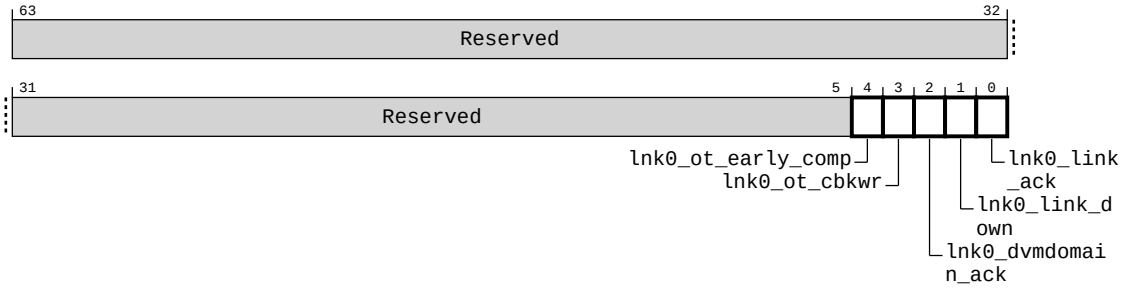


Table 4-70: por_ccg_ra_ccprtcl_link0_status attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4]	lnk0_ot_early_comp	Set, if there is an outstanding request for which early completion has been given for CCIX Link0	RO	1'b0
[3]	lnk0_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX Link0	RO	1'b0
[2]	lnk0_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for CCIX Link 0	RO	1'b0
[1]	lnk0_link_down	Link Down status; hardware updates this register bit to indicate Link Down status **1'b0** Link is not Down; hardware clears Link_DN when it receives a Link Up request **1'b1** Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	1'b1
[0]	lnk0_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request **1'b0** Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops granting protocol credits and starts returning protocol credits to the remote agent when Link_ACK is clear **1'b1** Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent NOTE: The local agent must clear Link_DN before setting Link_ACK.	RO	1'b0

4.3.3.20 por_ccg_ra_ccprtcl_link1_ctl

Functions as the CXRA CCIX Protocol Link 1 control register. Works with por_ccg_ra_ccprtcl_link1_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C10`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-55: por_ccg_ra_ccprtcl_link1_ctl

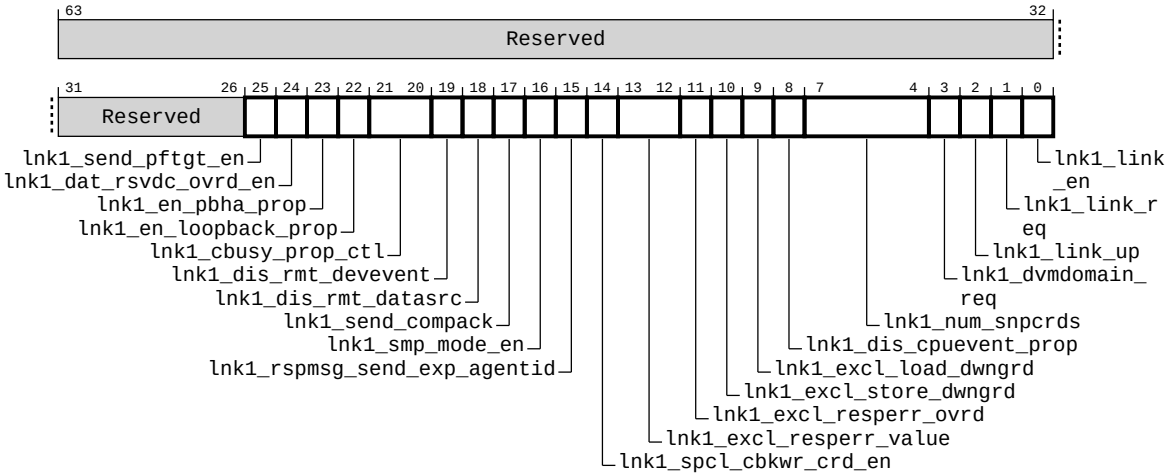


Table 4-71: por_ccg_ra_ccprtcl_link1_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	lnk1_send_pftgt_en	When set, enables sending Prefetch Target (CHI) over link 1. Note: This field is not-applicable in CXL mode for link1	RW	1'b1
[24]	lnk1_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 1. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[23]	lnk1_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 1.	RW	1'b1
[22]	lnk1_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 1.	RW	1'b1

Bits	Name	Description	Type	Reset
[21:20]	lnk1_cbusy_prop_ctl	Controls the propagation of Cbusy field for CCIX Link 1. **2'b00** Send RA Cbusy on all responses based on the limits programmed in por_ccg_ra_cbusy_limit_ctl **2'b01** Pass through remote CBusy on late completion responses (CompData, Comp) **2'b10** Greater of RA Cbusy or remote Cbusy. Applicable to responses where remote Cbusy can be sent NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	2'b0
[19]	lnk1_dis_rmt_devevent	When set, disables propagation of remote Dev Event field for CCIX Link 1. NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[18]	lnk1_dis_rmt_datasrc	When set, disables propagation of remote data source for CCIX Link 1. NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[17]	lnk1_send_compack	When set, sends CompAck for CCIX Link 1.	RW	1'b0
[16]	lnk1_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 1.	RW	Configuration dependent
[15]	lnk1_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 1	RW	1'b0
[14]	lnk1_spcl_cbkwr_crd_en	When set, notifies RA to use special credits from HA to send CopyBack writes on CCIX Link 1 NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[13:12]	lnk1_excl_resperr_value	Two bit value to override RespErr field of an exclusive response. Applicable only if lnk1_excl_resperr_ovrd bit is set. NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	2'b0
[11]	lnk1_excl_resperr_ovrd	When set, overrides the RespErr field of exclusive response with the lnk1_excl_resperr_value field NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[10]	lnk1_excl_store_dwngrd	When set, downgrades shareable exclusive store to shareable store when sending on CCIX Link 1 NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[9]	lnk1_excl_load_dwngrd	When set, downgrades shareable exclusive load to shareable load when sending on CCIX Link 1 NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[8]	lnk1_dis_cpuevent_prop	When set, disables the propagation of CPU Events on CCIX Link 1 NOTE: This field is applicable only when SMP Mode enable parameter is set.	RW	1'b0

Bits	Name	Description	Type	Reset
[7:4]	lnk1_num_snpcrds	Controls the number of CCIX snoop credits assigned to Link 1 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[3]	lnk1_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for CCIX Link 1	RW	1'b0
[2]	lnk1_link_up	Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent **1'b0** Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear **1'b1** Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent	RW	1'b0
[1]	lnk1_link_req	Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent **1'b0** Link Down request **1'b1** Link Up request The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.	RW	1'b0
[0]	lnk1_link_en	Enables CCIX Link 1 when set **1'b0** Link is disabled **1'b1** Link is enabled	RW	1'b0

4.3.3.21 por_ccg_ra_ccprtcl_link1_status

Functions as the CXRA CCIX Protocol Link 1 status register. Works with por_ccg_ra_ccprtcl_link1_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C18`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-56: por_ccg_ra_ccprtcl_link1_status

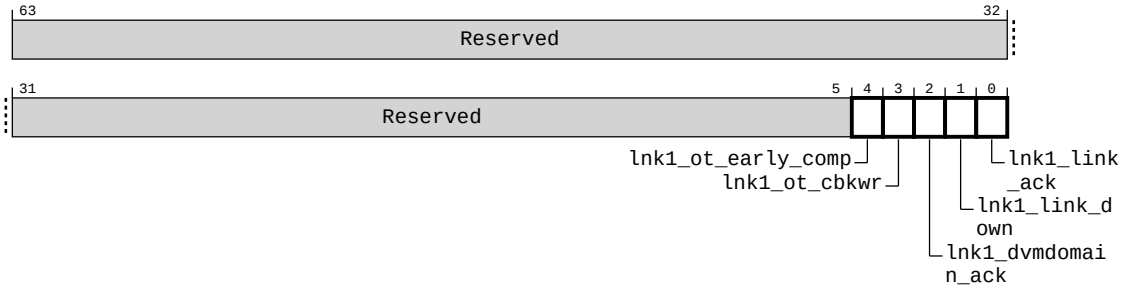


Table 4-72: por_ccg_ra_ccprtcl_link1_status attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4]	lnk1_ot_early_comp	Set, if there is an outstanding request for which early completion has been given for CCIX Link1	RO	1'b0
[3]	lnk1_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX Link1	RO	1'b0
[2]	lnk1_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for CCIX Link 1	RO	1'b0
[1]	lnk1_link_down	Link Down status; hardware updates this register bit to indicate Link Down status **1'b0** Link is not Down; hardware clears Link_DN when it receives a Link Up request **1'b1** Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	1'b1
[0]	lnk1_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request **1'b0** Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear **1'b1** Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent NOTE: The local agent must clear Link_DN before setting Link_ACK.	RO	1'b0

4.3.3.22 por_ccg_ra_ccprtcl_link2_ctl

Functions as the CXRA CCIX Protocol Link 2 control register. Works with por_ccg_ra_ccprtcl_link2_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C20

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-57: por_ccg_ra_ccprtcl_link2_ctl

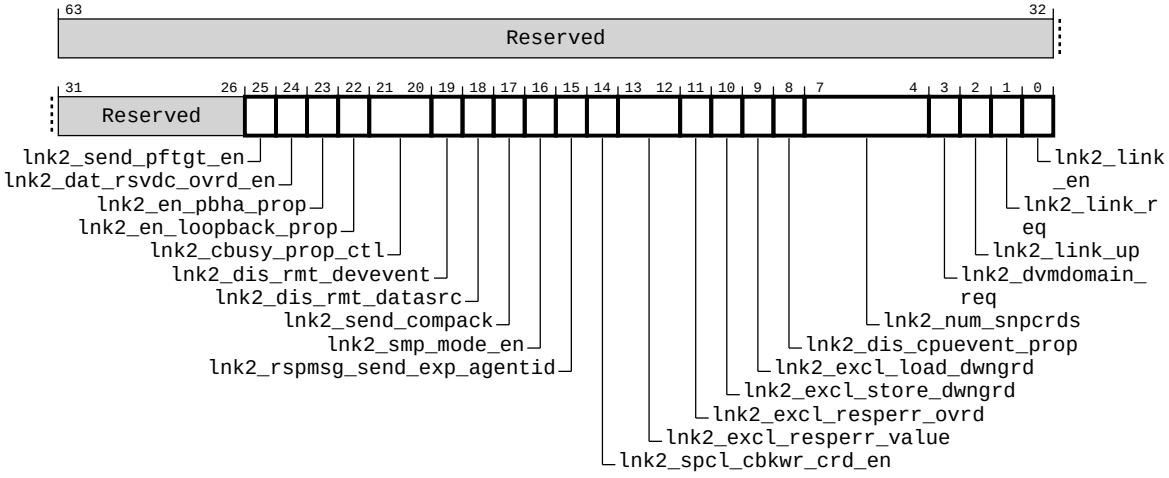


Table 4-73: por_ccg_ra_ccprtcl_link2_ctl attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[25]	lnk2_send_pftgt_en	When set, enables sending Prefetch Target (CHI) over link 2. Note: This field is not-applicable in CXL mode for link2	RW	1'b1
[24]	lnk2_dat_rsvdc_ovrd_en	When set, overrides CHI DAT RSVDC field with dat rsvdc strap input for incoming data on CCIX Link 2. Note: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode)	RW	1'b0
[23]	lnk2_en_pbha_prop	When set, enables propagation of PBHA on CCIX Link 2.	RW	1'b1
[22]	lnk2_en_loopback_prop	When set, enables propagation of LoopBack on CCIX Link 2.	RW	1'b1
[21:20]	lnk2_cbusy_prop_ctl	Controls the propagation of Cbusy field for CCIX Link 2. **2'b00** Send RA Cbusy on all responses based on the limits programmed in por_ccg_ra_cbusy_limit_ctl **2'b01** Pass through remote CBusy on late completion responses (CompData, Comp) **2'b10** Greater of RA Cbusy or remote Cbusy. Applicable to responses where remote Cbusy can be sent NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	2'b0
[19]	lnk2_dis_rmt_devevent	When set, disables propagation of remote Dev Event field for CCIX Link 2. NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[18]	lnk2_dis_rmt_datasrc	When set, disables propagation of remote data source for CCIX Link 2. NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[17]	lnk2_send_compack	When set, sends CompAck for CCIX Link 2.	RW	1'b0
[16]	lnk2_smp_mode_en	When set, enables Symmetric Multiprocessor Mode (SMP) Mode for CCIX Link 2.	RW	Configuration dependent
[15]	lnk2_rspmsg_send_exp_agentid	When set sends Expanded Agent ID on CCIX Response Messages for CCIX Link 2	RW	1'b0
[14]	lnk2_spcl_cbkwr_crd_en	When set, notifies RA to use special credits from HA to send CopyBack writes on CCIX Link 2 NOTE: This field is applicable only if the link programmed for SMP mode (i.e. SMP Mode enable bit is set)	RW	1'b0
[13:12]	lnk2_excl_resperr_value	Two bit value to override RespErr field of an exclusive response. Applicable only if lnk2_excl_resperr_ovrd bit is set. NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	2'b0
[11]	lnk2_excl_resperr_ovrd	When set, overrides the RespErr field of exclusive response with the lnk2_excl_resperr_value field NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[10]	lnk2_excl_store_dwngrd	When set, downgrades shareable exclusive store to shareable store when sending on CCIX Link 2 NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0
[9]	lnk2_excl_load_dwngrd	When set, downgrades shareable exclusive load to shareable load when sending on CCIX Link 2 NOTE: This field is applicable only when SMP Mode enable bit is clear (i.e. Non-SMP mode) and must only be used if the corresponding link end pair does not support exclusive monitoring	RW	1'b0

Bits	Name	Description	Type	Reset
[8]	lnk2_dis_cpuevent_prop	When set, disables the propagation of CPU Events on CCIX Link 2 NOTE: This field is applicable only when SMP Mode enable parameter is set.	RW	1'b0
[7:4]	lnk2_num_snpcrds	Controls the number of CCIX snoop credits assigned to Link 2 **4'h0** Total credits are equally divided across all links **4'h1** 25% of credits assigned **4'h2** 50% of credits assigned **4'h3** 75% of credits assigned **4'h4** 100% of credits assigned **4'hF** 0% of credits assigned	RW	4'b0
[3]	lnk2_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for CCIX Link 2	RW	1'b0
[2]	lnk2_link_up	Link Up status. Software writes this register bit to indicate Link status after polling Link_ACK and Link_DN status in the remote agent **1'b0** Link is not Up. Software clears Link_UP when Link_ACK status is clear and Link_DN status is set in both local and remote agents. The local agent stops responding to any protocol activity from remote agent, including acceptance of protocol credits, when Link_UP is clear **1'b1** Link is Up. Software sets Link_UP when Link_ACK status is set and Link_DN status is clear in both local and remote agents; the local agent starts sending local protocol credits to remote agent	RW	1'b0
[1]	lnk2_link_req	Link Up/Down request; software writes this register bit to request a Link Up or Link Down in the local agent **1'b0** Link Down request **1'b1** Link Up request The local agent does not return remote protocol credits yet since remote agent may still be in Link_UP state.	RW	1'b0
[0]	lnk2_link_en	Enables CCIX Link 2 when set **1'b0** Link is disabled **1'b1** Link is enabled	RW	1'b0

4.3.3.23 por_ccg_ra_ccprtcl_link2_status

Functions as the CXRA CCIX Protocol Link 2 status register. Works with por_ccg_ra_ccprtcl_link2_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C28

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-58: por_ccg_ra_ccprtcl_link2_status

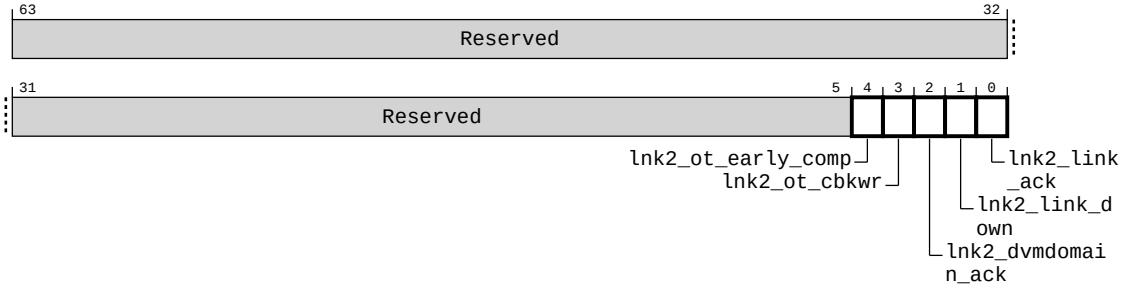


Table 4-74: por_ccg_ra_ccprtcl_link2_status attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4]	lnk2_ot_early_comp	Set, if there is an outstanding request for which early completion has been given for CCIX Link2	RO	1'b0
[3]	lnk2_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX Link2	RO	1'b0
[2]	lnk2_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for CCIX Link 2	RO	1'b0
[1]	lnk2_link_down	Link Down status; hardware updates this register bit to indicate Link Down status **1'b0** Link is not Down; hardware clears Link_DN when it receives a Link Up request **1'b1** Link is Down; hardware sets Link_DN after the local agent has received all local protocol credits. The local agent must continue to respond to any remote protocol activity, including accepting and returning remote protocol credits until Link Up is clear	RO	1'b1

Bits	Name	Description	Type	Reset
[0]	lnk2_link_ack	Link Up/Down acknowledge; hardware updates this register bit to acknowledge the software link request **1'b0** Link Down acknowledge; hardware clears Link_ACK on receiving a Link Down request; the local agent stops sending protocol credits to the remote agent when Link_ACK is clear **1'b1** Link Up acknowledge; hardware sets Link_ACK when the local agent is ready to start accepting protocol credits from the remote agent NOTE: The local agent must clear Link_DN before setting Link_ACK.	RO	1'b0

4.3.4 CCLA register descriptions

This section lists the CCLA registers.

4.3.4.1 por_ccla_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-59: por_ccla_node_info

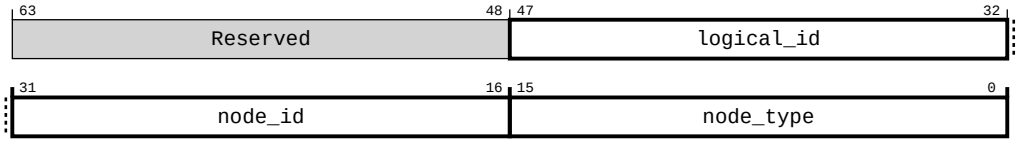


Table 4-75: por_ccla_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0105

4.3.4.2 por_ccla_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-60: por_ccla_child_info

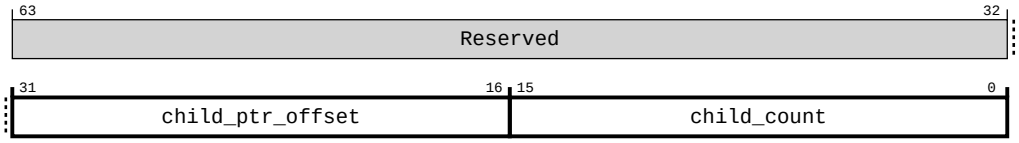


Table 4-76: por_ccla_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.4.3 por_ccla_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h988`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-61: por_ccla_secure_register_groups_override

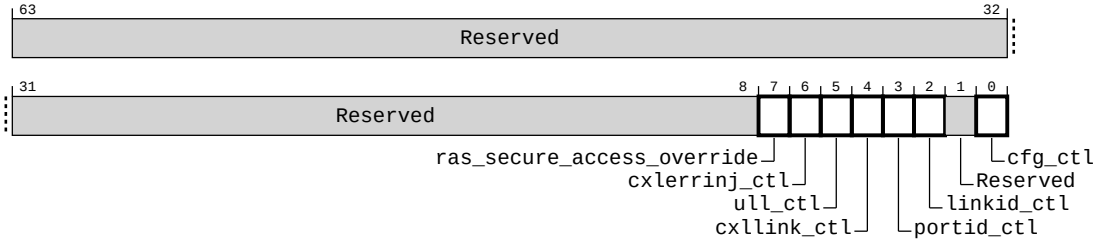


Table 4-77: por_ccla_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7]	ras_secure_access_override	Allow Non-secure access to Secure RAS registers	RW	1'b0
[6]	cxlerrinj_ctl	Allows Non-secure access to Secure CXL error injection registers	RW	1'b0
[5]	ull_ctl	Allows Non-secure access to Secure upper link layer control registers	RW	1'b0
[4]	cxllink_ctl	Allows Non-secure access to Secure CXL link layer registers	RW	1'b0
[3]	portid_ctl	Allows Non-secure access to Secure LA Port ID registers	RW	1'b0
[2]	linkid_ctl	Allows Non-secure access to Secure LA Link ID registers	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register	RW	1'b0

4.3.4.4 por_ccla_unit_info

Provides component identification information for CCLA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h910`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-62: por_ccla_unit_info

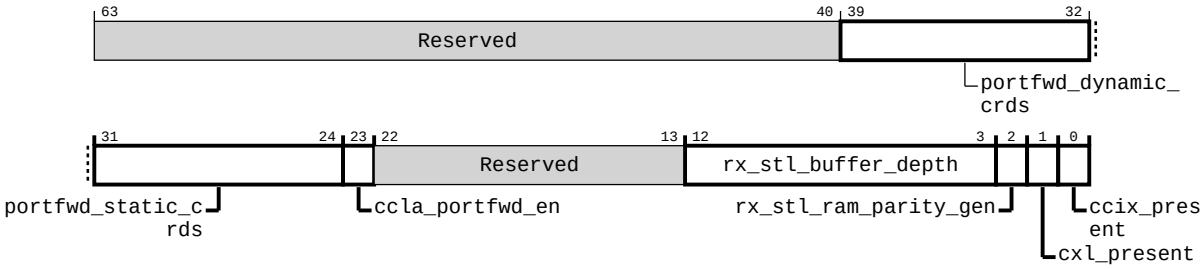


Table 4-78: por_ccla_unit_info attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	portfwd_dynamic_crds	Number of dynamic credits granted by this CCLA port for port forwarded traffic	RO	Configuration dependent
[31:24]	portfwd_static_crds	Number of static credits granted by this CCLA port for port forwarded traffic	RO	Configuration dependent
[23]	ccla_portfwd_en	Port forwarding is enabled at this CCLA port	RO	Configuration dependent
[22:13]	Reserved	Reserved	RO	-
[12:3]	rx_stl_buffer_depth	Depth of CCL stalling channel RX buffer for CXS RSP with data messages	RO	Configuration dependent
[2]	rx_stl_ram_parity_gen	Option to generate parity bits for the RX STL buffer	RO	Configuration dependent
[1]	cxl_present	Option to generate CXL support over CXS	RO	Configuration dependent
[0]	ccix_present	Option to generate CCIX support over CXS	RO	Configuration dependent

4.3.4.5 por_ccla_cfg_ctl

Functions as the configuration control register for CCLA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hB00

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-63: por_ccla_cfg_ctl

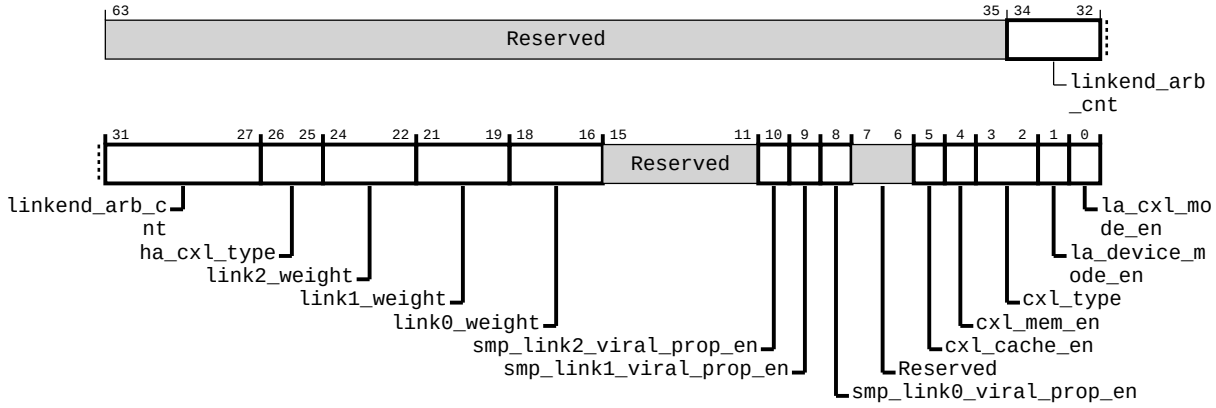


Table 4-79: por_ccla_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34:27]	linkend_arb_cnt	The count for how long each linkend is selected during linkend arbitration. If linkend_arb_cnt=8, each linkend is active for 8 cycles before switching to the next linkend	RW	8'h10
[26:25]	ha_cxl_type	Used to program CXL Type for HA **2'b00** Reserved **2'b01** Type1 **2'b10** Type2 **2'b11** Type3	RW	2'b01
[24:22]	link2_weight	Determines weight of link2 in CCL linkend arbitration	RW	3'b001
[21:19]	link1_weight	Determines weight of link1 in CCL linkend arbitration	RW	3'b001
[18:16]	link0_weight	Determines weight of link0 in CCL linkend arbitration	RW	3'b001

Bits	Name	Description	Type	Reset
[15:11]	Reserved	Reserved	RO	-
[10]	smp_link2_viral_prop_en	When set, enables viral propagation on SMP link2	RW	1'b0
[9]	smp_link1_viral_prop_en	When set, enables viral propagation on SMP link1	RW	1'b0
[8]	smp_link0_viral_prop_en	When set, enables viral propagation on SMP link0	RW	1'b0
[7:6]	Reserved	Reserved	RO	-
[5]	cxl_cache_en	Enable CXL .cache mode, by default is disabled	RW	1'b0
[4]	cxl_mem_en	Enable CXL .mem mode, by default is enabled	RW	1'b1
[3:2]	cxl_type	Used to program CXL Type for RA **2'b00** Reserved **2'b01** Type1 **2'b10** Type2 **2'b11** Type3	RW	2'b11
[1]	la_device_mode_en	Enable the Device mode, by default set to Host mode	RW	1'b0
[0]	la_cxl_mode_en	When set enables CXL mode	RW	1'b0

4.3.4.6 por_ccla_aux_ctl

Functions as the auxiliary control register for CCLA.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hB08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-64: por_ccla_aux_ctl

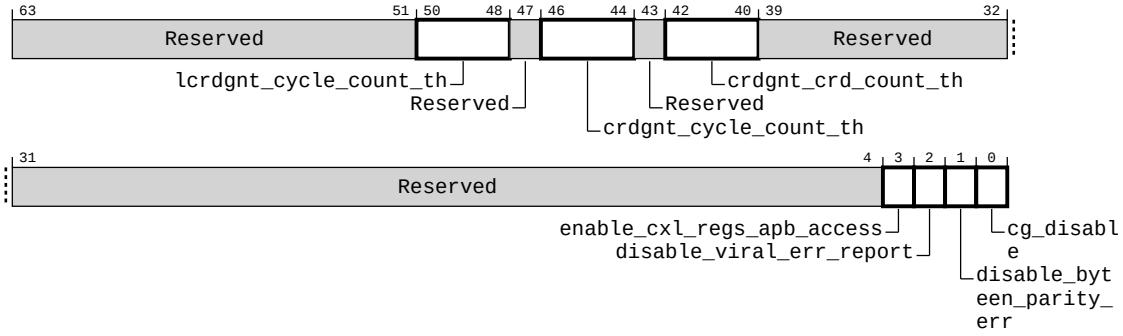


Table 4-80: por_ccla_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:51]	Reserved	Reserved	RO	-
[50:48]	lcrdgnt_cycle_count_th	Maximum number of cycles that need to be elapsed since previous piggyback credits were sent, to send a link credit grant message **3'b000** 8 cycles **3'b001** 16 cycles **3'b010** 32 cycles **3'b011** 64 cycles	RW	3'b001
[47]	Reserved	Reserved	RO	-
[46:44]	crdgnt_cycle_count_th	Maximum number of cycles that need to be elapsed since previous piggyback credits were sent, to send a protocol (Req, Dat, Snp..) credit grant message **3'b000** 32 cycles **3'b001** 64 cycles **3'b010** 128 cycles **3'b011** 256 cycles	RW	3'b010
[43]	Reserved	Reserved	RO	-
[42:40]	crdgnt_crd_count_th	Maximum number of protocol credits (i.e. Req, Dat, Snp..) that need to be accumulated to send a credit grant message **3'b000** 16 cycles **3'b001** 32 cycles **3'b010** 64 cycles **3'b011** 128 cycles	RW	3'b010
[39:4]	Reserved	Reserved	RO	-
[3]	enable_cxl_regs_apb_access	When set, Enables the APB access to the CXL registers and Disables the CMN access	RW	1'b0
[2]	disable_viral_err_report	When set, disables viral error reporting through CMN's error reporting mechanism	RW	1'b1
[1]	disable_byteen_parity_err	Disables CCLA RX RAM byte enable parity errors	RW	1'b0
[0]	cg_disable	Disables CCLA architectural clock gates	RW	1'b0

4.3.4.7 por_ccla_ccix_prop_capabilities

Contains CCIX-supported properties.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-65: por_ccla_ccix_prop_capabilities

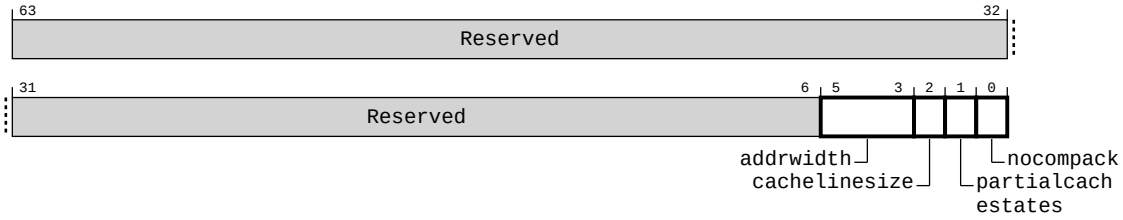


Table 4-81: por_ccla_ccix_prop_capabilities attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:3]	addrwidth	Address width supported **3'b000** 48b **3'b001** 52b **3'b010** 56b **3'b011** 60b **3'b100** 64b	RO	3'b001

Bits	Name	Description	Type	Reset
[2]	cachelinesize	Cacheline size supported **1'b0** 64B **1'b1** 128B	RO	1'b0
[1]	partialcachestates	Partial cache states supported **1'b0** False **1'b1** True	RO	1'b0
[0]	nocompack	No CompAck supported **1'b0** False **1'b1** True	RO	1'b1

4.3.4.8 por_ccla_cxs_attr_capabilities

Contains CXS supported attributes.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC08`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-66: por_ccla_cxs_attr_capabilities

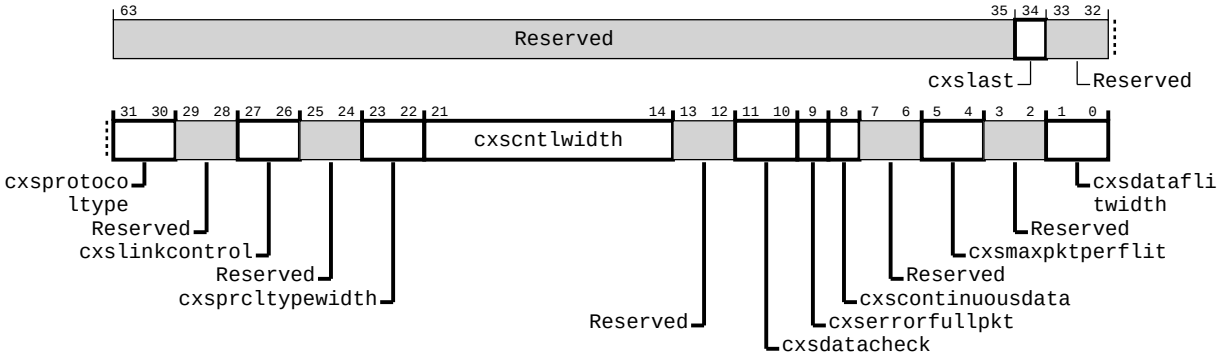


Table 4-82: por_ccla_cxs_attr_capabilities attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34]	cxslast	CXS LAST signal is supported	RO	Configuration dependent
[33:32]	Reserved	Reserved	RO	-
[31:30]	cxsprotocoltype	CXS Protocol type signal is supported	RO	Configuration dependent
[29:28]	Reserved	Reserved	RO	-
[27:26]	cxslinkcontrol	Set to Explicit Credit Return.	RO	Configuration dependent
[25:24]	Reserved	Reserved	RO	-
[23:22]	cxsprcltypewidth	Width of CXS TX/RX control	RO	Configuration dependent
[21:14]	cxscntlwidth	Width of CXS TX/RX control	RO	Configuration dependent
[13:12]	Reserved	Reserved	RO	-
[11:10]	cxsdatacheck	CXS datacheck supported **2'b00** None **2'b01** Parity **2'b10** SECDED	RO	Configuration dependent
[9]	cxserrorfullpkt	CXS error full packet supported **1'b0** False **1'b1** True	RO	Configuration dependent
[8]	cxscontinuousdata	CXS continuous data supported **1'b0** False **1'b1** True	RO	Configuration dependent
[7:6]	Reserved	Reserved	RO	-
[5:4]	cxsmaxpktperflit	CXS maximum packets per flit supported **2'b00** 2 **2'b01** 3 **2'b10** 4	RO	Configuration dependent
[3:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1:0]	cxsdataflitwidth	CXS data flit width supported **2'b00** 256b **2'b01** 512b **2'b10** 1024b	RO	2'b01

4.3.4.9 por_ccla_permsg_pyld_0_63

Contains bits[63:0] of CCIX Protocol Error (PER) Message payload.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD00`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-67: por_ccla_permsg_pyld_0_63

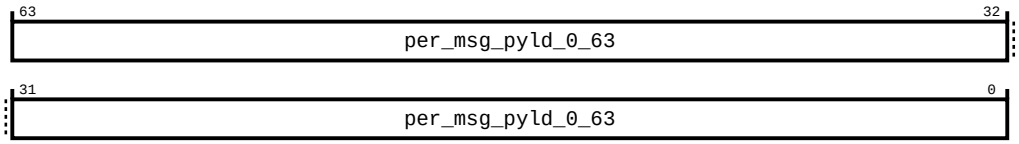


Table 4-83: por_ccla_permsg_pyld_0_63 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_0_63	Protocol Error Msg Payload[63:0]	RW	64'b0

4.3.4.10 por_ccla_permsg_pyld_64_127

Contains bits[127:64] of CCIX Protocol Error (PER) Message payload.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD08

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-68: por_ccla_permsg_pyld_64_127

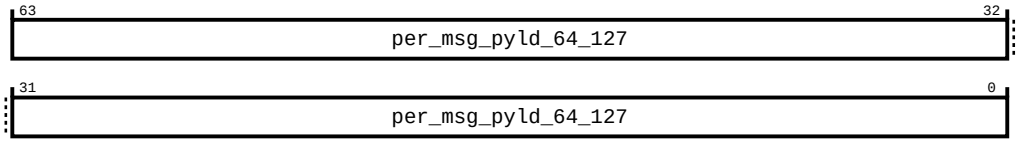


Table 4-84: por_ccla_permsg_pyld_64_127 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_64_127	Protocol Error Msg Payload[127:64]	RW	64'b0

4.3.4.11 por_ccla_permsg_pyld_128_191

Contains bits[192:128] of CCIX Protocol Error (PER) Message payload.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD10`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-69: por_ccla_permsg_pyld_128_191

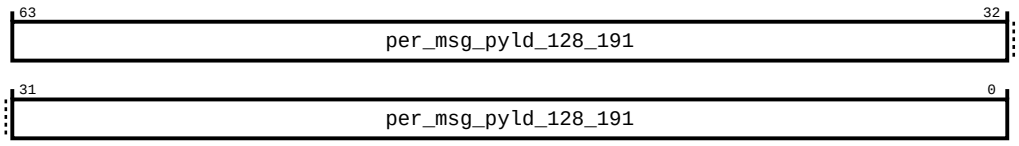


Table 4-85: por_ccla_permsg_pyld_128_191 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_128_191	Protocol Error Msg Payload[191:128]	RW	64'b0

4.3.4.12 por_ccla_permsg_pyld_192_255

Contains bits[255:192] of CCIX Protocol Error (PER) Message payload.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD18`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-70: por_ccla_permsg_pyld_192_255

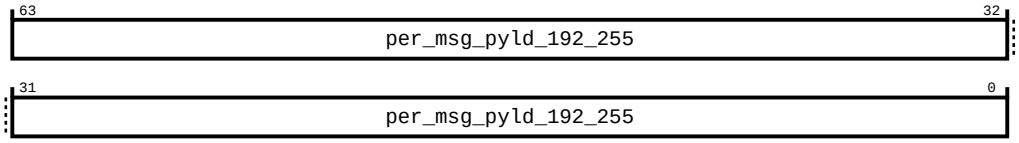


Table 4-86: por_ccla_permsg_pyld_192_255 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_192_255	Protocol Error Msg Payload[255:192]	RW	64'b0

4.3.4.13 por_ccla_permsg_ctl

Contains Control bits to trigger CCIX Protocol Error (PER) Message.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD20`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-71: por_ccla_permsg_ctl

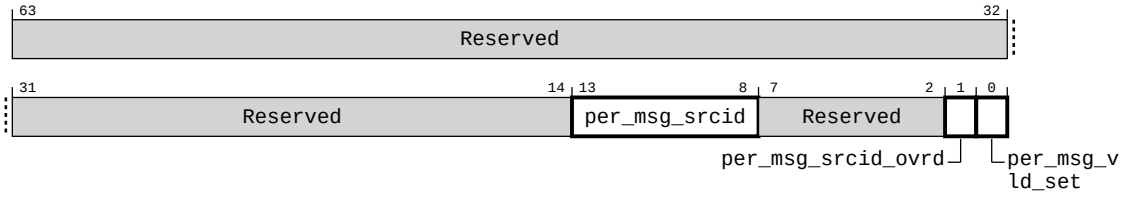


Table 4-87: por_ccla_permsg_ctl attributes

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13:8]	per_msg_srcid	Contains Source ID used on CCIX Protocol Error Msg. Used when per_msg_srcid_ovrd is set.	RW	6'b0
[7:2]	Reserved	Reserved	RO	-
[1]	per_msg_srcid_ovrd	When set, overrides the Source ID on Protocol Error Msg by value specified in this register. Or else the source ID from payload[55:48] is used.	RW	1'b0
[0]	per_msg_vld_set	When set, sends CCIX Protocol Error Msg. Must be cleared after the current error is processed and before a new error message is triggered	RW	1'b0

4.3.4.14 por_ccla_err_agent_id

Contains Error Agent ID. Must be programmed by CCIX discovery s/w. Used as TargetID on CCIX Protocol Error (PER) Message.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD28`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-72: por_ccla_err_agent_id

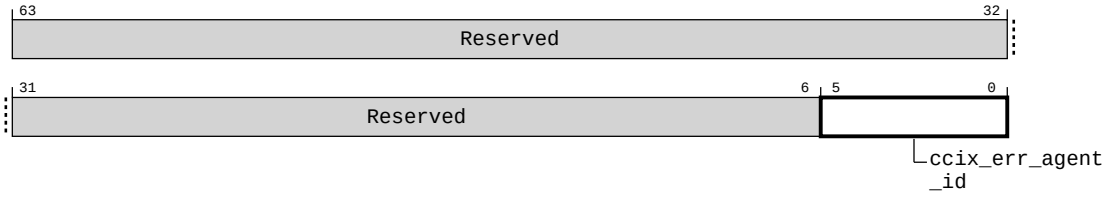


Table 4-88: por_ccla_err_agent_id attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:0]	ccix_err_agent_id	CCIX Error AgentID	RW	6'b0

4.3.4.15 por_ccla_agentid_to_portid_reg0

Specifies the mapping of Agent ID to Port ID for Agent IDs 0 to 7.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD30`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-73: por_ccla_agentid_to_portid_reg0

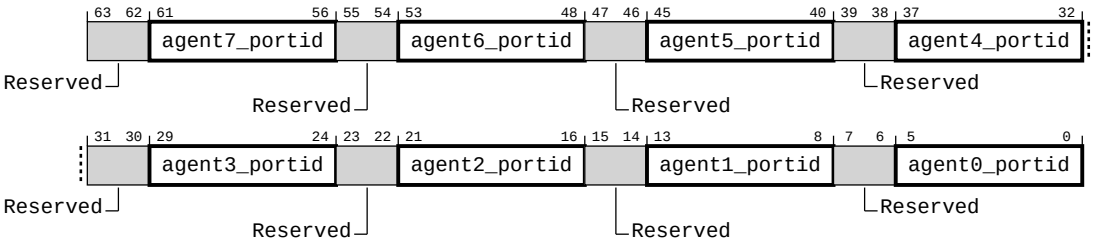


Table 4-89: por_ccla_agentid_to_portid_reg0 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent7_portid	Specifies the Port ID for Agent ID 7	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent6_portid	Specifies the Port ID for Agent ID 6	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent5_portid	Specifies the Port ID for Agent ID 5	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent4_portid	Specifies the Port ID for Agent ID 4	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent3_portid	Specifies the Port ID for Agent ID 3	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent2_portid	Specifies the Port ID for Agent ID 2	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent1_portid	Specifies the Port ID for Agent ID 1	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent0_portid	Specifies the Port ID for Agent ID 0	RW	6'h0

4.3.4.16 por_ccla_agentid_to_portid_reg1

Specifies the mapping of Agent ID to Port ID for Agent IDs 8 to 15.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD38`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-74: por_ccla_agentid_to_portid_reg1

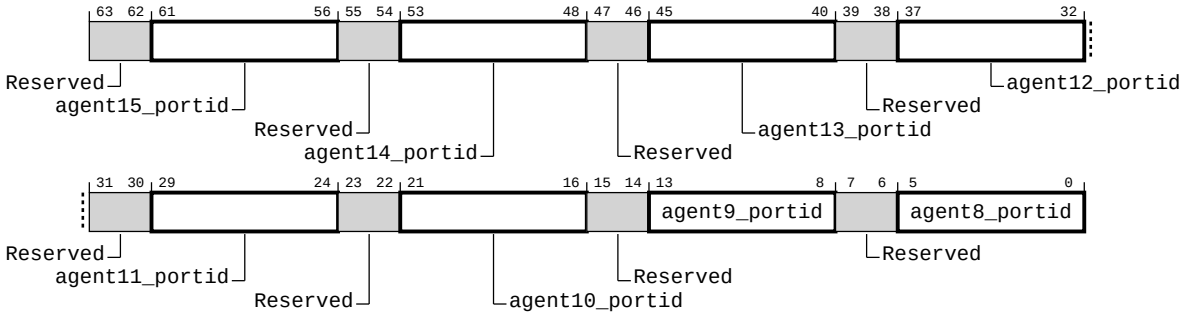


Table 4-90: por_ccla_agentid_to_portid_reg1 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent15_portid	Specifies the Port ID for Agent ID 15	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent14_portid	Specifies the Port ID for Agent ID 14	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent13_portid	Specifies the Port ID for Agent ID 13	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent12_portid	Specifies the Port ID for Agent ID 12	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent11_portid	Specifies the Port ID for Agent ID 11	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent10_portid	Specifies the Port ID for Agent ID 10	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent9_portid	Specifies the Port ID for Agent ID 9	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent8_portid	Specifies the Port ID for Agent ID 8	RW	6'h0

4.3.4.17 por_ccla_agentid_to_portid_reg2

Specifies the mapping of Agent ID to Port ID for Agent IDs 16 to 23.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD40`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-75: por_ccla_agentid_to_portid_reg2

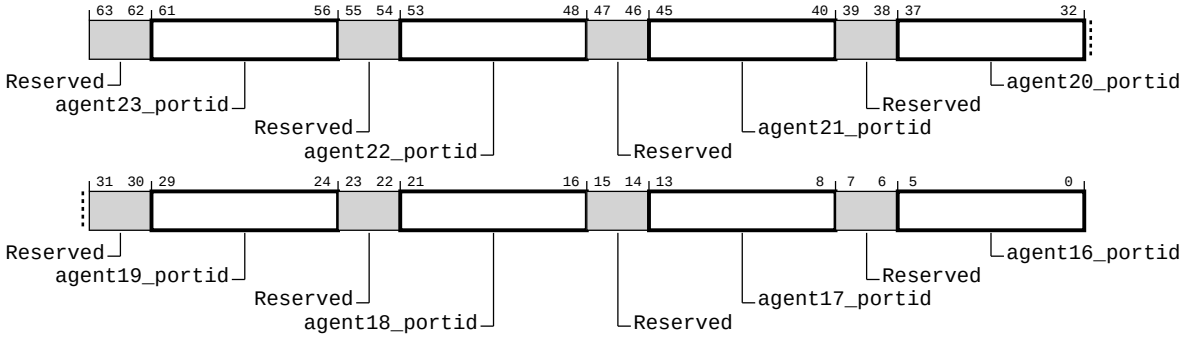


Table 4-91: por_ccla_agentid_to_portid_reg2 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent23_portid	Specifies the Port ID for Agent ID 23	RW	6'h0
[55:54]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[53:48]	agent22_portid	Specifies the Port ID for Agent ID 22	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent21_portid	Specifies the Port ID for Agent ID 21	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent20_portid	Specifies the Port ID for Agent ID 20	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent19_portid	Specifies the Port ID for Agent ID 19	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent18_portid	Specifies the Port ID for Agent ID 18	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent17_portid	Specifies the Port ID for Agent ID 17	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent16_portid	Specifies the Port ID for Agent ID 16	RW	6'h0

4.3.4.18 por_ccla_agentid_to_portid_reg3

Specifies the mapping of Agent ID to Port ID for Agent IDs 24 to 31.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD48`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-76: por_ccla_agentid_to_portid_reg3

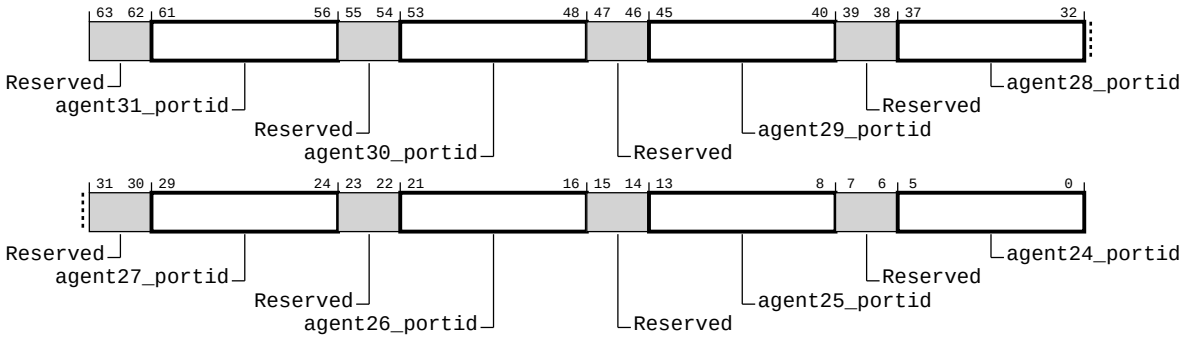


Table 4-92: por_ccla_agentid_to_portid_reg3 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent31_portid	Specifies the Port ID for Agent ID 31	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent30_portid	Specifies the Port ID for Agent ID 30	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent29_portid	Specifies the Port ID for Agent ID 29	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent28_portid	Specifies the Port ID for Agent ID 28	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent27_portid	Specifies the Port ID for Agent ID 27	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent26_portid	Specifies the Port ID for Agent ID 26	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent25_portid	Specifies the Port ID for Agent ID 25	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent24_portid	Specifies the Port ID for Agent ID 24	RW	6'h0

4.3.4.19 por_ccla_agentid_to_portid_reg4

Specifies the mapping of Agent ID to Port ID for Agent IDs 32 to 39.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD50

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-77: por_ccla_agentid_to_portid_reg4

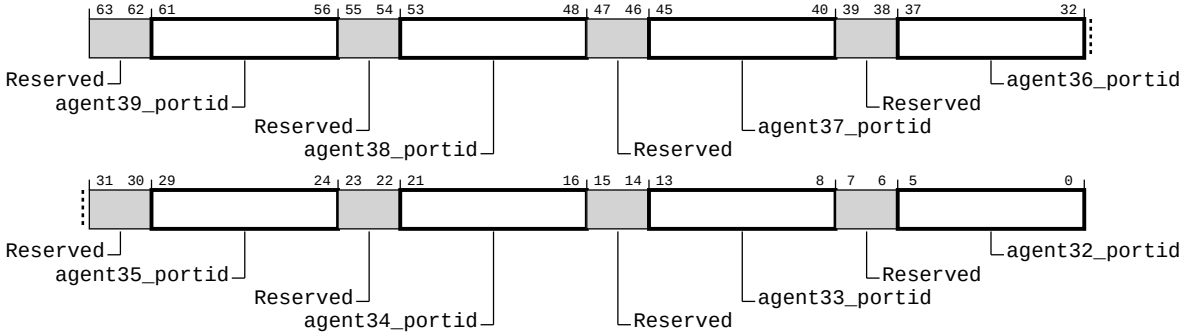


Table 4-93: por_ccla_agentid_to_portid_reg4 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent39_portid	Specifies the Port ID for Agent ID 39	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent38_portid	Specifies the Port ID for Agent ID 38	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent37_portid	Specifies the Port ID for Agent ID 37	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent36_portid	Specifies the Port ID for Agent ID 36	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent35_portid	Specifies the Port ID for Agent ID 35	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent34_portid	Specifies the Port ID for Agent ID 34	RW	6'h0
[15:14]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[13:8]	agent33_portid	Specifies the Port ID for Agent ID 33	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent32_portid	Specifies the Port ID for Agent ID 32	RW	6'h0

4.3.4.20 por_ccla_agentid_to_portid_reg5

Specifies the mapping of Agent ID to Port ID for Agent IDs 40 to 47.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD58`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-78: por_ccla_agentid_to_portid_reg5

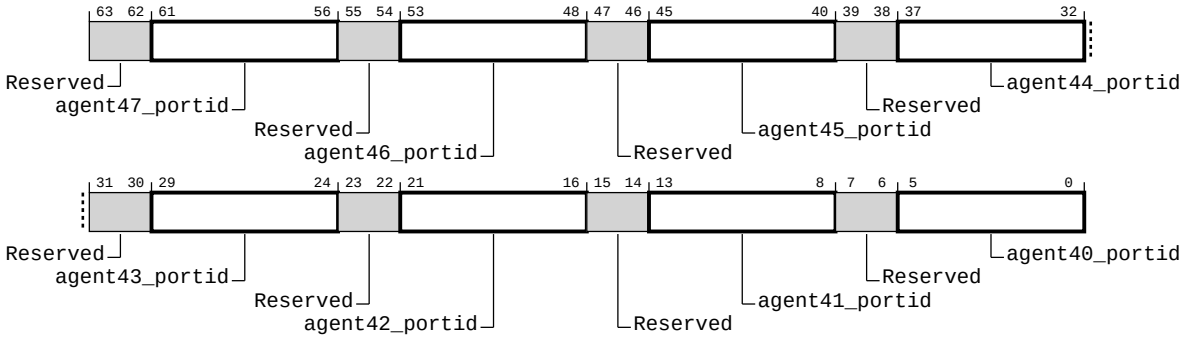


Table 4-94: por_ccla_agentid_to_portid_reg5 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent47_portid	Specifies the Port ID for Agent ID 47	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent46_portid	Specifies the Port ID for Agent ID 46	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent45_portid	Specifies the Port ID for Agent ID 45	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent44_portid	Specifies the Port ID for Agent ID 44	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent43_portid	Specifies the Port ID for Agent ID 43	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent42_portid	Specifies the Port ID for Agent ID 42	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent41_portid	Specifies the Port ID for Agent ID 41	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent40_portid	Specifies the Port ID for Agent ID 40	RW	6'h0

4.3.4.21 por_ccla_agentid_to_portid_reg6

Specifies the mapping of Agent ID to Port ID for Agent IDs 48 to 55.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD60`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-79: por_ccla_agentid_to_portid_reg6

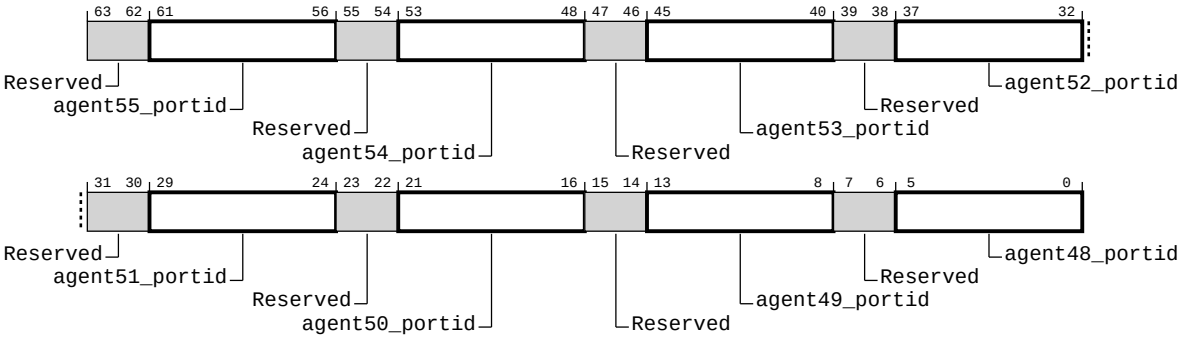


Table 4-95: por_ccla_agentid_to_portid_reg6 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent55_portid	Specifies the Port ID for Agent ID 55	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent54_portid	Specifies the Port ID for Agent ID 54	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent53_portid	Specifies the Port ID for Agent ID 53	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent52_portid	Specifies the Port ID for Agent ID 52	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent51_portid	Specifies the Port ID for Agent ID 51	RW	6'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent50_portid	Specifies the Port ID for Agent ID 50	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent49_portid	Specifies the Port ID for Agent ID 49	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent48_portid	Specifies the Port ID for Agent ID 48	RW	5'h0

4.3.4.22 por_ccla_agentid_to_portid_reg7

Specifies the mapping of Agent ID to Port ID for Agent IDs 56 to 63.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD68`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-80: por_ccla_agentid_to_portid_reg7

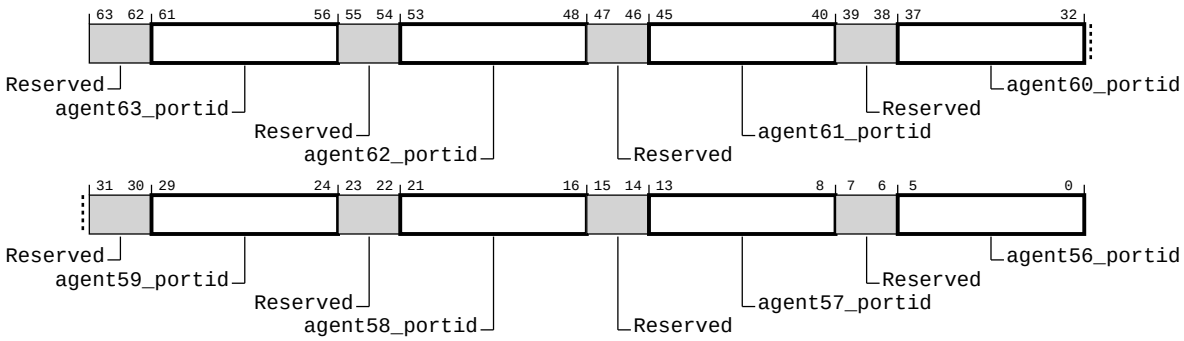


Table 4-96: por_ccla_agentid_to_portid_reg7 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	agent63_portid	Specifies the Port ID for Agent ID 63	RW	6'h0
[55:54]	Reserved	Reserved	RO	-
[53:48]	agent62_portid	Specifies the Port ID for Agent ID 62	RW	6'h0
[47:46]	Reserved	Reserved	RO	-
[45:40]	agent61_portid	Specifies the Port ID for Agent ID 61	RW	6'h0
[39:38]	Reserved	Reserved	RO	-
[37:32]	agent60_portid	Specifies the Port ID for Agent ID 60	RW	6'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	agent59_portid	Specifies the Port ID for Agent ID 59	RW	6'h0

Bits	Name	Description	Type	Reset
[23:22]	Reserved	Reserved	RO	-
[21:16]	agent58_portid	Specifies the Port ID for Agent ID 58	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	agent57_portid	Specifies the Port ID for Agent ID 57	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	agent56_portid	Specifies the Port ID for Agent ID 56	RW	6'h0

4.3.4.23 por_ccla_agentid_to_portid_val

Specifies which Agent ID to Port ID mappings are valid.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD70`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-81: por_ccla_agentid_to_portid_val

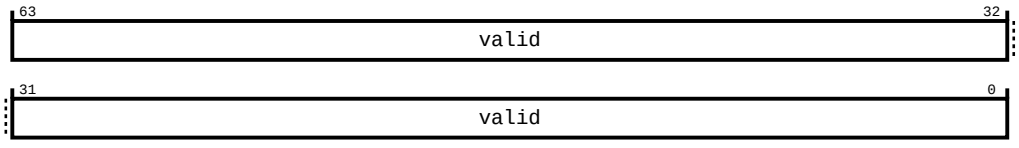


Table 4-97: por_ccla_agentid_to_portid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Port ID is valid; bit number corresponds to logical Agent ID number (from 0 to 63)	RW	63'h0

4.3.4.24 por_ccla_portfwd_en

Functions as the Port-to-Port forwarding control register. Works with por_ccla_portfwd_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD78`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-82: por_ccla_portfwd_en

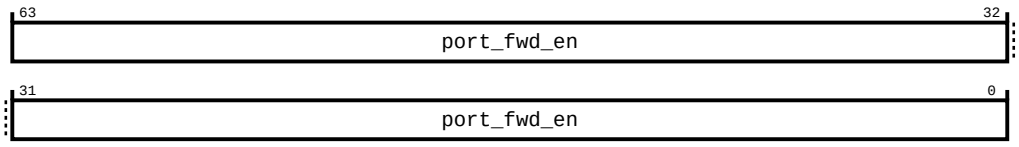


Table 4-98: por_ccla_portfwd_en attributes

Bits	Name	Description	Type	Reset
[63:0]	port_fwd_en	Bit vector, where each bit represents logical PortID of a Port (CCG) present on CMN. Each bit when set, enables Port-to-Port forwarding with the corresponding port. **1'b0** Port-to-Port forwarding is disabled **1'b1** Port-to-Port forwarding is enabled	RW	64'b0

4.3.4.25 por_ccla_portfwd_status

Functions as the Port-to-Port forwarding status register. Works with por_ccla_portfwd_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD80`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-83: por_ccla_portfwd_status

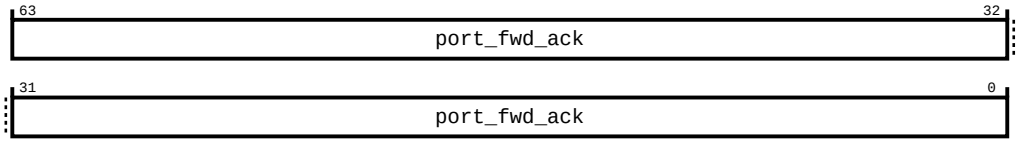


Table 4-99: por_ccla_portfwd_status attributes

Bits	Name	Description	Type	Reset
[63:0]	port_fwd_ack	Bit vector, where each bit represents logical PortID of a Port (CCG) present on CMN. Each bit represents the status of the port-to-port control request sent to the corresponding port. **1'b0** Port-to-Port forwarding channel is de-active. **1'b1** Port-to-Port forwarding channel is active	RO	64'b0

4.3.4.26 por_ccla_portfwd_req

Functions as the Port-to-Port forwarding control register. Works with por_ccla_portfwd_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD88`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-84: por_ccla_portfwd_req

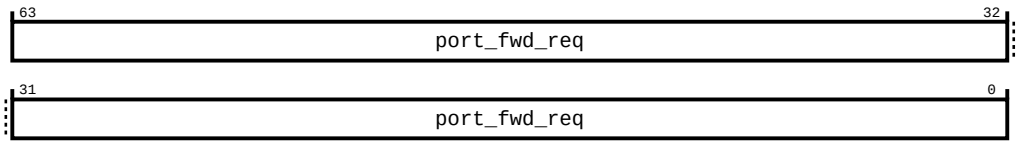


Table 4-100: por_ccla_portfwd_req attributes

Bits	Name	Description	Type	Reset
[63:0]	port_fwd_req	Bit vector, where each bit represents logical PortID of a Port (CCG) present on CMN. Each bit is used to control the communication channel with the corresponding port. **1'b0** Port-to-Port forwarding channel de-activate request **1'b1** Port-to-Port forwarding channel activate request	RW	64'b0

4.3.4.27 por_ccla_linkid_to_hops

Specifies number of portforward hops for the linkid.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD90

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-85: por_ccla_linkid_to_hops

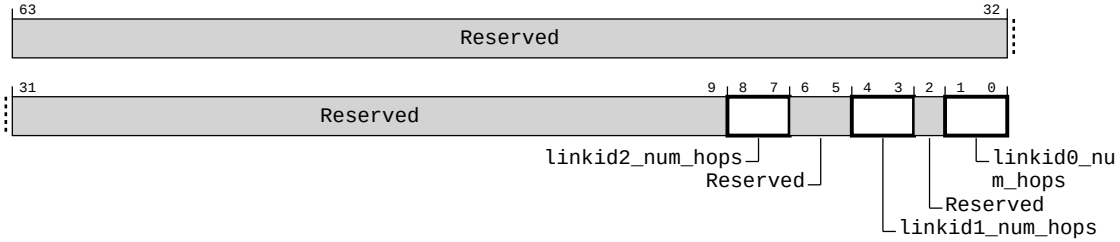


Table 4-101: por_ccla_linkid_to_hops attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8:7]	linkid2_num_hops	Specifies the number of portforward hops for linkid2	RW	2'b00
[6:5]	Reserved	Reserved	RO	-
[4:3]	linkid1_num_hops	Specifies the number of portforward hops for linkid1	RW	2'b00
[2]	Reserved	Reserved	RO	-
[1:0]	linkid0_num_hops	Specifies the number of portforward hops for linkid0	RW	2'b00

4.3.4.28 por_ccla_cxl_link_rx_credit_ctl

CXL Link Rx Credit Control Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-86: por_ccla_cxl_link_rx_credit_ctl

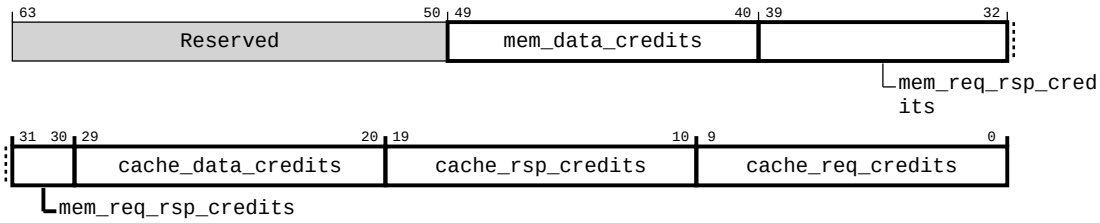


Table 4-102: por_ccla_cxl_link_rx_credit_ctl attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Credits to advertise for Mem Data channel at init	RW	Configuration dependent
[39:30]	mem_req_rsp_credits	Credits to advertise for Mem Request or Response channel at init	RW	Configuration dependent
[29:20]	cache_data_credits	Credits to advertise for Cache Data channel at init	RW	Configuration dependent
[19:10]	cache_rsp_credits	Credits to advertise for Cache Response channel at init	RW	Configuration dependent
[9:0]	cache_req_credits	Credits to advertise for Cache Request channel at init	RW	Configuration dependent

4.3.4.29 por_ccla_cxl_link_rx_credit_return_stat

CXL Link Rx Credit Return Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE08

Type

RO

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-87: por_ccla_cxl_link_rx_credit_return_stat

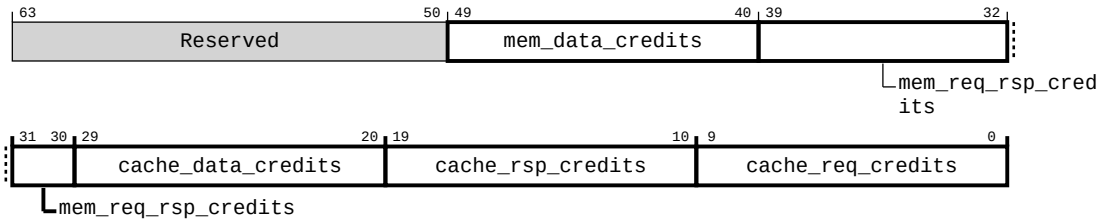


Table 4-103: por_ccla_cxl_link_rx_credit_return_stat attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits to be returned	RO	10'b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits to be returned	RO	10'b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits to be returned	RO	10'b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits to be returned	RO	10'b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits to be returned	RO	10'b0

4.3.4.30 por_ccla_cxl_link_tx_credit_stat

CXL Link Tx Credit Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE10

Type

RO

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-88: por_ccla_cxl_link_tx_credit_stat

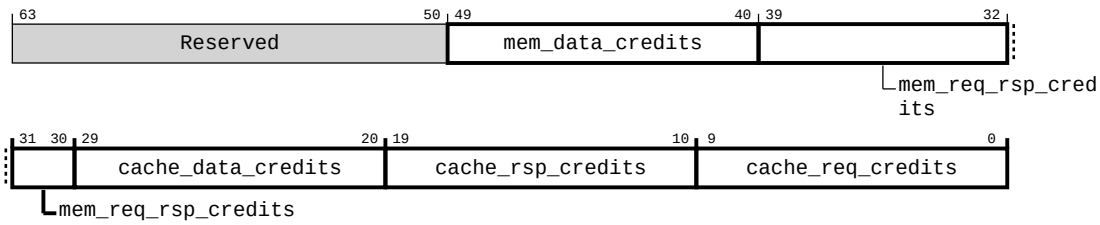


Table 4-104: por_ccla_cxl_link_tx_credit_stat attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits for TX	RO	10'b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits for TX	RO	10'b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits for TX	RO	10'b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits for TX	RO	10'b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits for TX	RO	10'b0

4.3.4.31 por_ccla_cxl_security_policy

Contains CXL Security Policy

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE50`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-89: por_ccla_cxl_security_policy

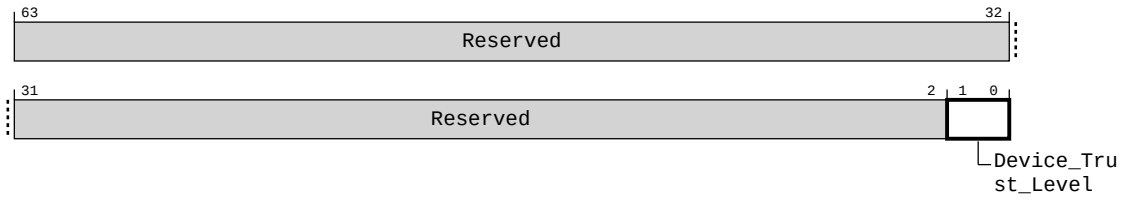


Table 4-105: por_ccla_cxl_security_policy attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	Device_Trust_Level	0 --> Trusted CXL Device. At this setting a CXL Device will be able to get access on CXL.cache for both host-attached and device attached memory ranges. The Host can still protect security sensitive memory regions. '1 --> Trusted for Device Attached Memory Range Only. At this setting a CXL Device will be able to get access on CXL.cache for device attached memory ranges only. Requests on CXL.cache for host-attached memory ranges will be aborted by the Host. '2 --> Untrusted CXL Device. At this setting all requests on CXL.cache will be aborted by the Host. Please note that these settings only apply to requests on CXL.cache. The device can still source requests on CXL.io regardless of these settings. Protection on CXL.io will be implemented using IOMMU based page tables. Default value of this field is 2.	RW	2'h2

4.3.4.32 por_ccla_cxl_hdm_decoder_capability

Contains CXL_HDM_Decoder_Capability_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE78

Type

RO

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-90: por_ccla_cxl_hdm_decoder_capability

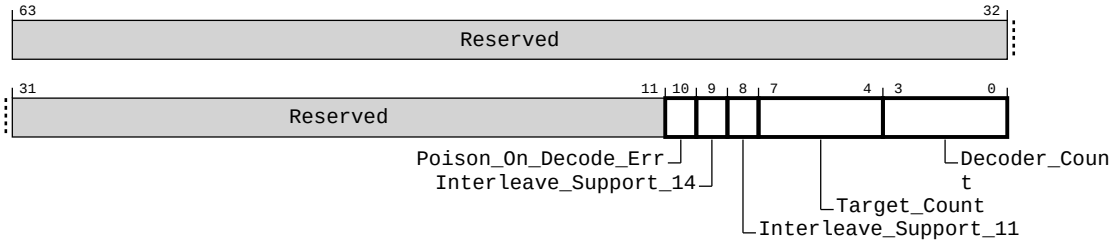


Table 4-106: por_ccla_cxl_hdm_decoder_capability attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	Poison_On_Decode_Err	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	1'h0

Bits	Name	Description	Type	Reset
[9]	Interleave_Support_14	If set the component supports interleaving based on Address bit 14 Address bit 13 and Address bit 12. Root ports and switches shall always set this bit indicating support for interleaving based on Address bits 14-12.	RO	1'h0
[8]	Interleave_Support_11	If set the component supports interleaving based on Address bit 11 Address bit 10 Address bit 9 and Address bit 8. Root Ports and Upstream Switch Ports shall always set this bit indicating support for interleaving based on Address bit 11-8.	RO	1'h0
[7:4]	Target_Count	The number of target ports each decoder supports (applicable to Upstream Switch Port and Root Port only). Maximum of 8. **1** 1 target port **2** 2 target ports **8** 8 target ports **Other** Reserved	RO	4'h1
[3:0]	Decoder_Count	Reports the number of memory address decoders implemented by the component. **0** 1 Decoder **1** 2 Decoders **2** 4 Decoders **3** 6 Decoders **4** 8 Decoders **5** 10 Decoders **Others** Reserved	RO	4'h0

4.3.4.33 por_ccla_cxl_hdm_decoder_global_control

Contains CXL_HDM_Decoder_Global_Control_Register . Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE80`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-91: por_ccla_cxl_hdm_decoder_global_control

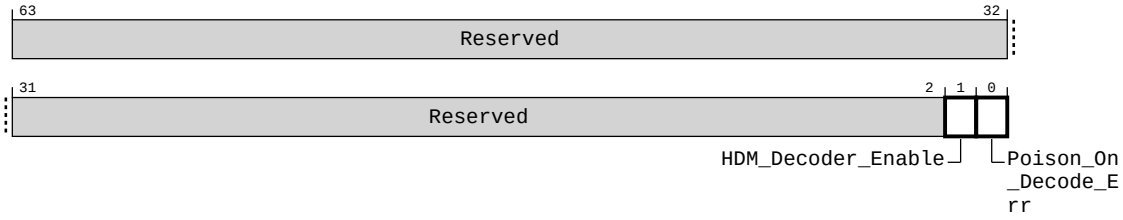


Table 4-107: por_ccla_cxl_hdm_decoder_global_control attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	HDM_Decoder_Enable	This bit is only applicable to CXL.mem devices and shall return 0 on Root Ports and Upstream Switch Ports. When this bit is set device shall use HDM decoders to decode CXL.mem transactions and not use HDM Base registers in DVSEC ID 0. Root Ports and Upstream Switch Ports always use HDM Decoders to decode CXL.mem transactions.	RW	1'h0
[0]	Poison_On_Decode_Err	This bit is RO and is hard-wired to 0 if Poison On Decode Error Capability=0. If set the component returns poison on read access to addresses that are not positively decoded by the component. If clear the component returns all 1s data without a poison under such scenarios.	RW	1'h0

4.3.4.34 por_ccla_cxl_hdm_decoder_0_base_low

Contains CXL_HDM_Decoder_0_Base_High_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE88`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-92: por_ccla_cxl_hdm_decoder_0_base_low

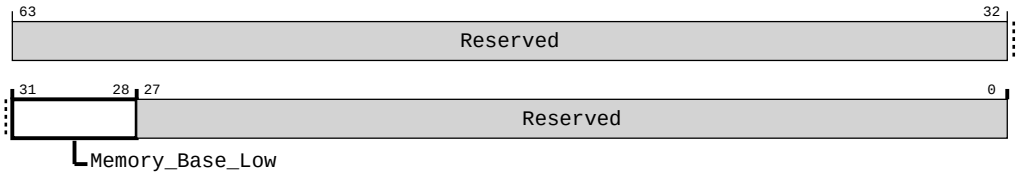


Table 4-108: por_ccla_cxl_hdm_decoder_0_base_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of the base of the address range managed by decoder 0	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.4.35 por_ccla_cxl_hdm_decoder_0_base_high

Contains CXL_HDM_Decoder_0_Base_Low_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE90`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-93: por_ccla_cxl_hdm_decoder_0_base_high

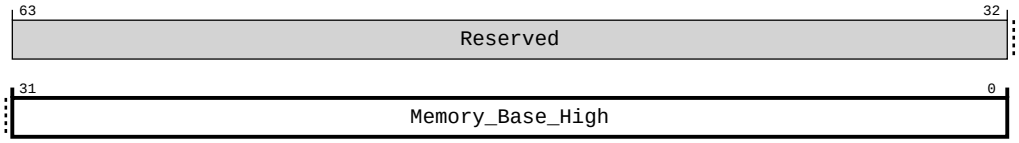


Table 4-109: por_ccla_cxl_hdm_decoder_0_base_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of the base of the address range managed by decoder 0.	RWL	32'h0

4.3.4.36 por_ccla_cxl_hdm_decoder_0_size_low

Contains CXL_HDM_Decoder_0_Size_Low_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE98`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-94: por_ccla_cxl_hdm_decoder_0_size_low

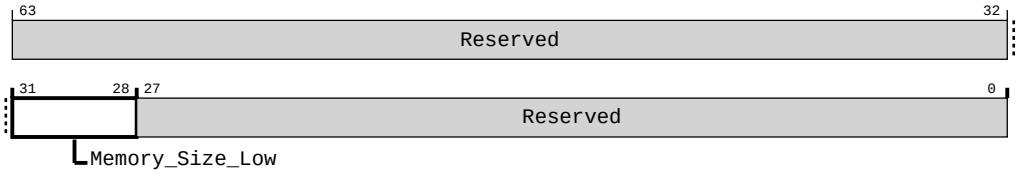


Table 4-110: por_ccla_cxl_hdm_decoder_0_size_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Size_Low	Corresponds to bits 31:28 of the size of the address range managed by decoder 0	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.4.37 por_ccla_cxl_hdm_decoder_0_size_high

Contains CXL_HDM_Decoder_0_Size_High_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEA0`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-95: por_ccla_cxl_hdm_decoder_0_size_high

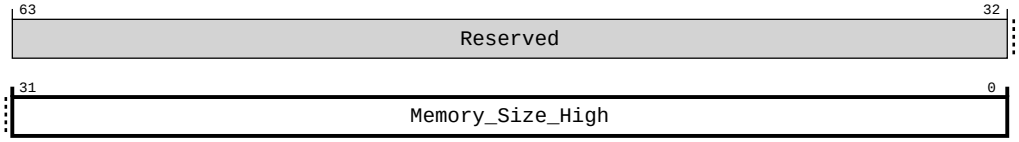


Table 4-111: por_ccla_cxl_hdm_decoder_0_size_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Size_High	Corresponds to bits 63:32 of the size of address range managed by decoder 0.	RWL	32'h0

4.3.4.38 por_ccla_cxl_hdm_decoder_0_control

Contains CXL_HDM_Decoder_0_Control_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEA8`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-96: por_ccla_cxl_hdm_decoder_0_control

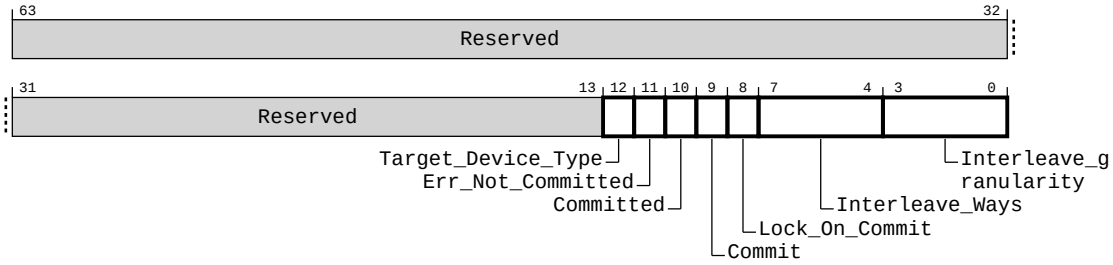


Table 4-112: por_ccla_cxl_hdm_decoder_0_control attributes

Bits	Name	Description	Type	Reset
[63:13]	Reserved	Reserved	RO	-
[12]	Target_Device_Type	**0** Target is a CXL Type 2 Device **1** Target is a CXL Type 3 Device	RWL	1'h0
[11]	Err_Not_Committed	Indicates the decode programming had an error and decoder is not active.	RWL	1'h0
[10]	Committed	Indicates the decoder is active	RWL	1'h0
[9]	Commit	Software sets this to 1 to commit this decoder	RWL	1'h0
[8]	Lock_On_Commit	If set all RWL fields in Decoder 0 registers will become read only when Committed changes to 1.	RWL	1'h0
[7:4]	Interleave_Ways	The number of targets across which this memory range is interleaved. 0 - 1 way 1 - 2 way 2 4 way 3 8 way All other reserved	RWL	4'h0
[3:0]	Interleave_granularity	The number of consecutive bytes that are assigned to each target in the Target List. **0** 256 Bytes **1** 512 Bytes **2** 1024 Bytes (1KB) **3** 2048 Bytes (2KB) **4** 4096 Bytes (4KB) **5** 8192 Bytes (8 KB) **6** 16384 Bytes (16 KB)	RWL	4'h0

4.3.4.39 por_ccla_cxl_hdm_decoder_0_dpa_skip_low

Contains CXL_HDM_Decoder_0_DPA_Skip_Low_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEC0

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-97: por_ccla_cxl_hdm_decoder_0_dpa_skip_low

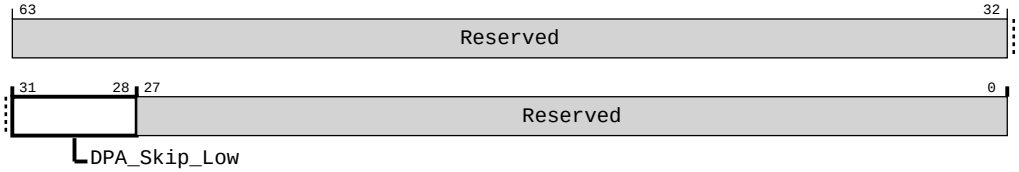


Table 4-113: por_ccla_cxl_hdm_decoder_0_dpa_skip_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	DPA_Skip_Low	Corresponds to bits 31:28 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.4.40 por_ccla_cxl_hdm_decoder_0_dpa_skip_high

Contains CXL_HDM_Decoder_0_DPA_Skip_High_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEC8

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-98: por_ccla_cxl_hdm_decoder_0_dpa_skip_high

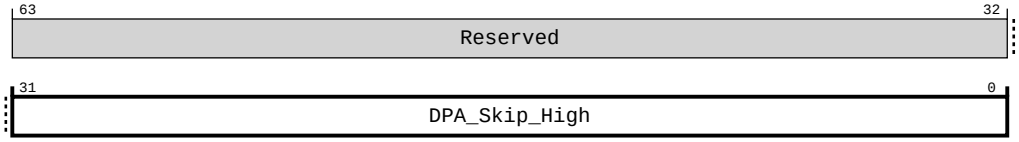


Table 4-114: por_ccla_cxl_hdm_decoder_0_dpa_skip_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	DPA_Skip_High	Corresponds to bits 63:32 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	32'h0

4.3.4.41 por_ccla_snoop_filter_group_id

Contains Snoop_Filter_Group_ID

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hED0

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-99: por_ccla_snoop_filter_group_id

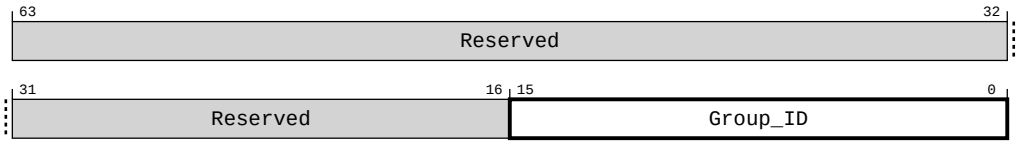


Table 4-115: por_ccla_snoop_filter_group_id attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	Group_ID	Uniquely identifies a snoop filter instance that is used to track CXL.cache devices below this Port. All Ports that share a single Snoop Filter instance shall set this field to the same value.	RW	16'h0

4.3.4.42 por_ccla_snoop_filter_effective_size

Contains Snoop_Filter_Effective_Size

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hED8`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-100: por_ccla_snoop_filter_effective_size

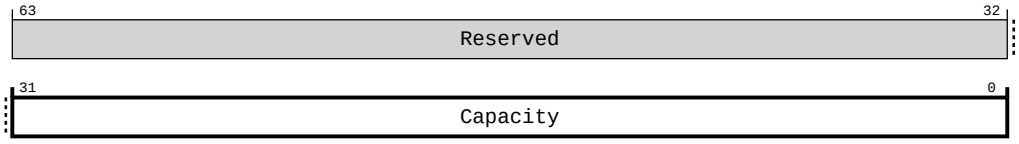


Table 4-116: por_ccla_snoop_filter_effective_size attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Capacity	Effective Snoop Filter Capacity representing the size of cache that can be effectively tracked by the Snoop Filter with this Group ID in multiples of 64K.	RW	32'h0

4.3.4.43 por_ccla_dvsec_cxl_range_1_base_high

Contains DVSEC_CXL_Range_1_Base_High. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEE0`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-101: por_ccla_dvsec_cxl_range_1_base_high

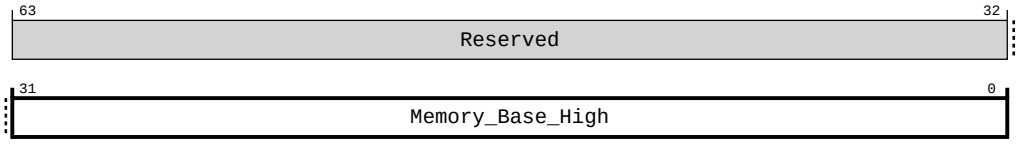


Table 4-117: por_ccla_dvsec_cxl_range_1_base_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder 0 Base High Register for backward compatibility reasons.	RWL	32'h0

4.3.4.44 por_ccla_dvsec_cxl_range_1_base_low

Contains DVSEC_CXL_Range_1_Base_Low. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEE8`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-102: por_ccla_dvsec_cxl_range_1_base_low

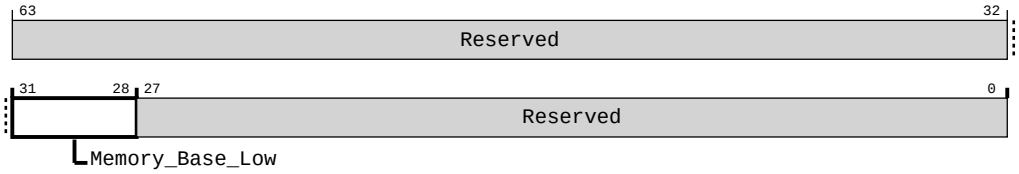


Table 4-118: por_ccla_dvsec_cxl_range_1_base_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder 0 Base Low Register for backward compatibility reasons.	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.4.45 por_ccla_dvsec_cxl_range_2_base_high

Contains DVSEC_CXL_Range_2_Base_High. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEF0`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-103: por_ccla_dvsec_cxl_range_2_base_high

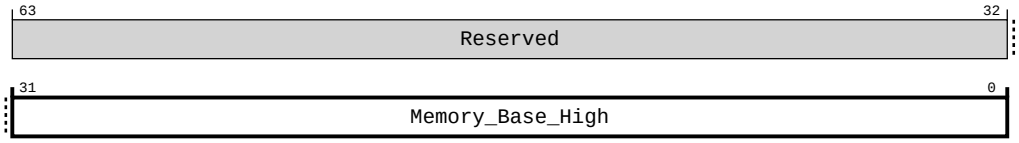


Table 4-119: por_ccla_dvsec_cxl_range_2_base_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match the corresponding CXL HDM Decoder Base High Register for backward compatibility reasons.	RWL	32'h0

4.3.4.46 por_ccla_dvsec_cxl_range_2_base_low

Contains DVSEC_CXL_Range_2_Base_Low. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEF8`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-104: por_ccla_dvsec_cxl_range_2_base_low

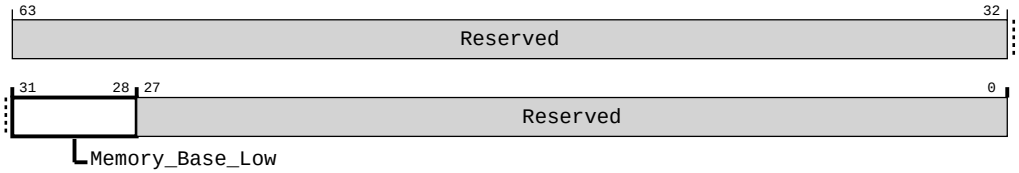


Table 4-120: por_ccla_dvsec_cxl_range_2_base_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.4.47 por_ccla_dvsec_cxl_control

Contains DVSEC_CXL_Control. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF00`

Type

RWL

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-105: por_ccla_dvsec_cxl_control

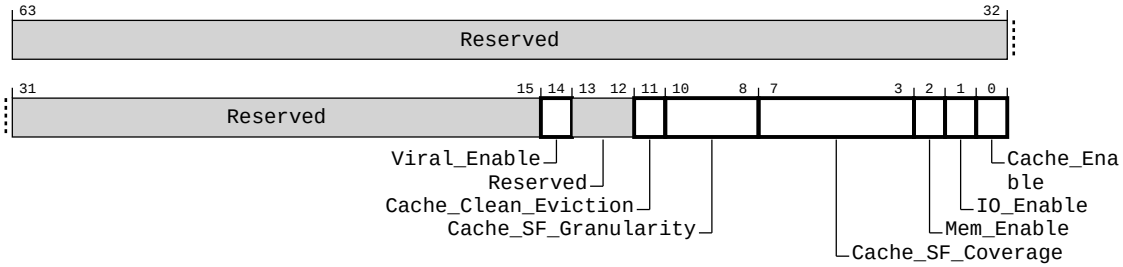


Table 4-121: por_ccla_dvsec_cxl_control attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14]	Viral_Enable	When set enables Viral handling in the CXL device. Locked by CONFIG_LOCK. If 0 the CXL device may ignore the viral that it receives	RWL	1'h0
[13:12]	Reserved	Reserved	RO	-
[11]	Cache_Clean_Eviction	Performance hint to the device. Locked by CONFIG_LOCK. **0** Indicates clean evictions from device caches are needed for best performance **1** Indicates clean evictions from device caches are NOT needed for best performance	RWL	1'h0
[10:8]	Cache_SF_Granularity	Performance hint to the device. Locked by CONFIG_LOCK. **000** Indicates 64B granular tracking on the Host **001** Indicates 128B granular tracking on the Host **010** Indicates 256B granular tracking on the Host **011** Indicates 512B granular tracking on the Host **100** Indicates 1KB granular tracking on the Host **101** Indicates 2KB granular tracking on the Host **110** Indicates 4KB granular tracking on the Host **111** Reserved	RWL	3'h0
[7:3]	Cache_SF_Coverage	Performance hint to the device. Locked by CONFIG_LOCK. 0x00: Indicates no Snoop Filter coverage on the Host For all other values of N: Indicates Snoop Filter coverage on the Host of 2^(N+15d) Bytes.	RWL	5'h0
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	1'h0
[1]	IO_Enable	When set enables CXL.io protocol operation when in Flex Bus.CXL mode.	RWL	1'h1
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	1'h0

4.3.4.48 por_ccla_dvsec_cxl_control2

Contains DVSEC_CXL_Control2. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF08

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-106: por_ccla_dvsec_cxl_control2

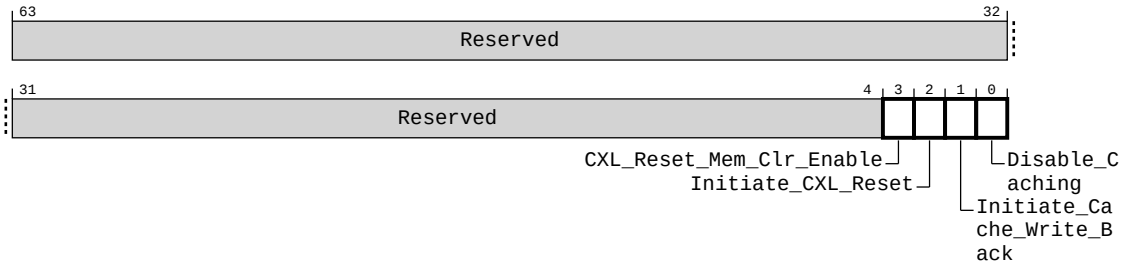


Table 4-122: por_ccla_dvsec_cxl_control2 attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	CXL_Reset_Mem_Clr_Enable	When set and CXL Reset Mem Clr Capable returns 1 Device shall clear or randomize volatile HDM ranges as part of the CXL Reset operation. When CXL Reset Mem Clr Capable is clear this bit is ignored and volatile HDM ranges may or may not be cleared or randomized during CXL Reset.	RW	1'h0
[2]	Initiate_CXL_Reset	When set to 1 device shall initiate CXL Reset as defined in Section 9.7. This bit always returns the value of 0 when read by the software. A write of 0 is ignored.	RW	1'h0

Bits	Name	Description	Type	Reset
[1]	Initiate_Cache_Write_Back	When set to 1 device shall write back all modified lines in the local cache and invalidate all lines. The device shall send CacheFlushed message to host as required by CXL.Cache protocol to indicate it does not hold any modified lines.	RW	1'h0
[0]	Disable_Caching	When set to 1 device shall no longer cache new modified lines in its local cache. Device shall continue to correctly respond to CXL.cache transactions.	RW	1'h0

4.3.4.49 por_ccla_dvsec_cxl_lock

Contains DVSEC_CXL_Lock. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF10`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-107: por_ccla_dvsec_cxl_lock

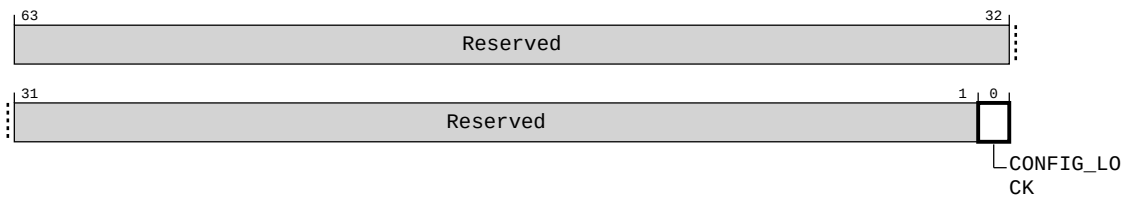


Table 4-123: por_ccla_dvsec_cxl_lock attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	CONFIG_LOCK	When set all register fields in the PCIe DVSEC for CXL Devices Capability with RWL attribute become read only. Consult individual register fields for details. This bit is cleared upon device Conventional Reset. This bit and all the fields that are locked by this bit are not affected by CXL Reset.	RW	1'h0

4.3.4.50 por_ccla_dvsec_flex_bus_port_control

Contains DVSEC_Flex_Bus_Port_Control

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF18`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-108: por_ccla_dvsec_flex_bus_port_control

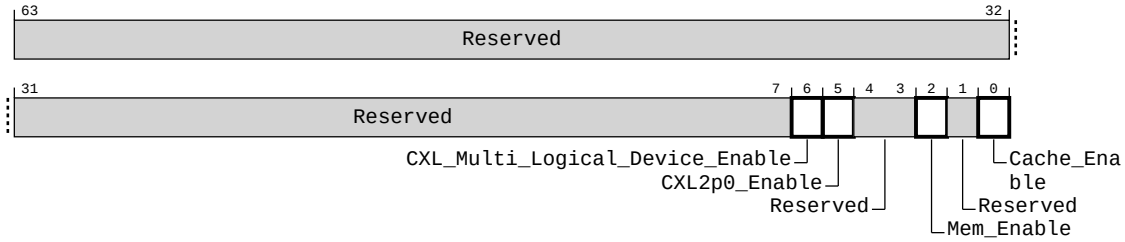


Table 4-124: por_ccla_dvsec_flex_bus_port_control attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	CXL_Multi_Logical_Device_Enable	When set enable Multi-Logical Device operation when in Flex Bus.CXL mode	RW	1'h0
[5]	CXL2p0_Enable	When set enable CXL2.0 protocol operation when in Flex Bus.CXL mode.	RW	1'h0
[4:3]	Reserved	Reserved	RO	-
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode.	RW	1'h0
[1]	Reserved	Reserved	RO	-
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode.	RW	1'h0

4.3.4.51 por_ccla_err_capabilities_control

Contains err_capabilities_control. Only applicable to device. Host doesn't use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF40`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-109: por_ccla_err_capabilities_control

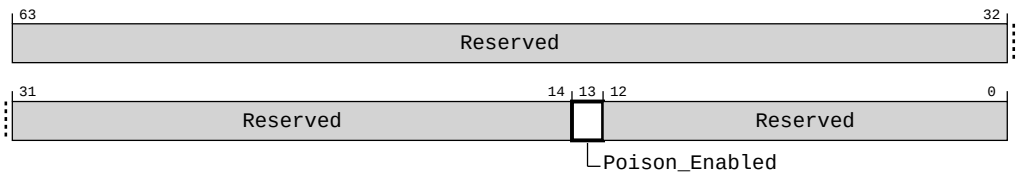


Table 4-125: por_ccla_err_capabilities_control attributes

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13]	Poison_Enabled	If this bit is 0 CXL 1.1 Upstream Ports CXL 1.1 Downstream Ports and CXL 2.0 Root Port shall treat poison received on CXL.cache or CXL.mem as uncorrectable error and log the error in Uncorrectable Error Status Register. If this bit is 1 these ports shall treat poison received on CXL.cache or CXL.mem as correctable error and log the error in Correctable Error Status Register. This bit defaults to 1. This bit is hardwired to 1 in CXL 2.0 Upstream Switch Port CXL 2.0 Downstream Switch Port and CXL 2.0 device.	RW	1'h0
[12:0]	Reserved	Reserved	RO	-

4.3.4.52 por_ccla_IDE_key_refresh_time_control

Contains IDE_key_refresh_time_control. Not applicable to CMN 700

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF58`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-110: por_ccla_IDE_key_refresh_time_control

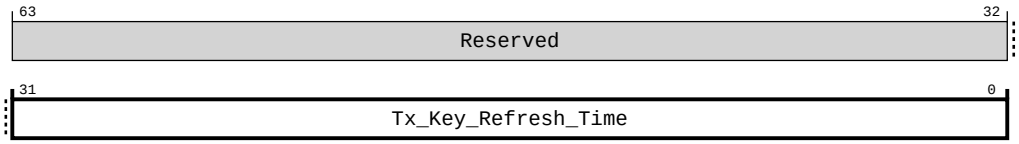


Table 4-126: por_ccla_IDE_key_refresh_time_control attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Tx_Key_Refresh_Time	Minimum number of flits transmitter needs to block transmission of protocol flits after IDE.Start has sent. Used when switching keys.	RW	32'h0

4.3.4.53 por_ccla_IDE_truncation_transmit_delay_control

Contains IDE_truncation_transmit_delay_control.Not applicable to CMN 700

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF60`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-111: por_ccla_IDE_truncation_transmit_delay_control

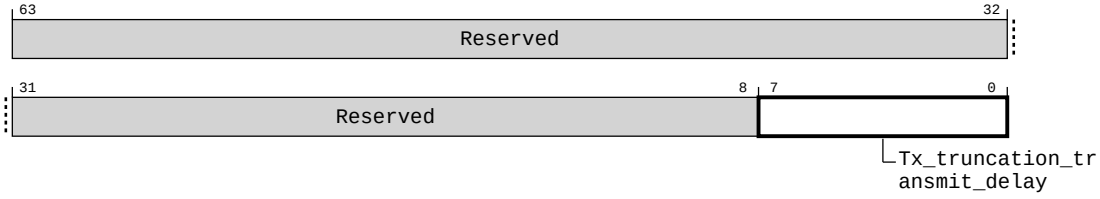


Table 4-127: por_ccla_IDE_truncation_transmit_delay_control attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7:0]	Tx_truncation_transmit_delay	This parameter feeds into the computation of minimum number of IDE idle flits Transmitter needs send after sending a truncated MAC flit.	RW	8'h0

4.3.4.54 por_ccla_ll_to_ull_msg

Contains lll_to_ull_message

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF70`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-112: por_ccla_ll_to_ull_msg

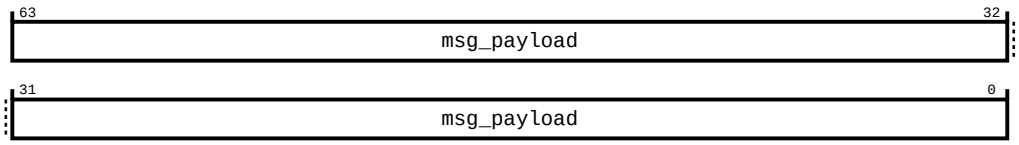


Table 4-128: por_ccla_ll_to_ull_msg attributes

Bits	Name	Description	Type	Reset
[63:0]	msg_payload	Contains 64 bits of message sent from ll to ull	RW	64'h0

4.3.4.55 por_ccla_cxl_timeout_isolation_control

Contains cxl_timeout_isolation_control. Not applicable to CMN 700

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF80

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-113: por_ccla_cxl_timeout_isolation_control

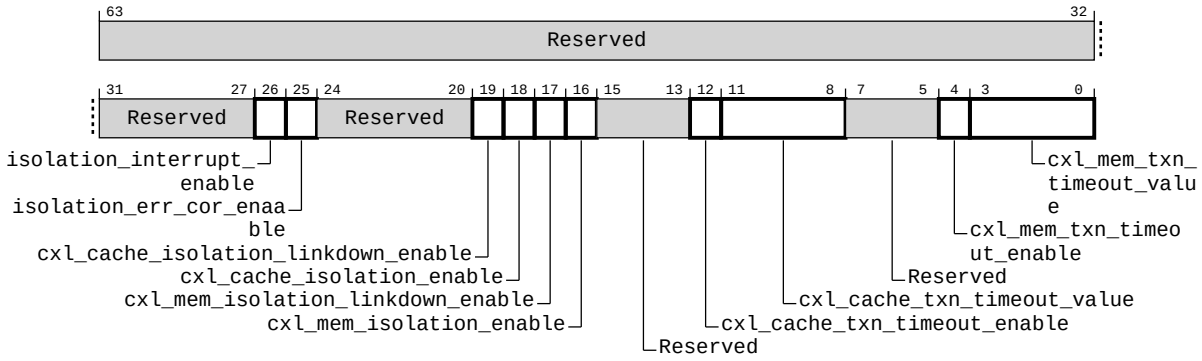


Table 4-129: por_ccla_cxl_timeout_isolation_control attributes

Bits	Name	Description	Type	Reset
[63:27]	Reserved	Reserved	RO	-
[26]	isolation_interrupt_enable	When Set this bit enables the generation of an interrupt to indicate that Isolation has been triggered.	RW	1'h0

Bits	Name	Description	Type	Reset
[25]	isolation_err_cor_enaable	When Set this bit enables the sending of an ERR_COR Message to indicate Isolation has been triggered.	RW	1'h0
[24:20]	Reserved	Reserved	RO	-
[19]	cxl_cache_isolation_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.cache enters Isolation mode.	RW	1'h0
[18]	cxl_cache_isolation_enable	This field allows System Software to enable CXL.cache Isolation actions.	RW	1'h0
[17]	cxl_mem_isolation_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.mem enters Isolation mode.	RW	1'h0
[16]	cxl_mem_isolation_enable	This field allows System Software to enable CXL.mem Isolation actions. If this field is set Isolation actions will be triggered if either a CXL.mem Transaction Timeout is detected or if the CXL link went down.	RW	1'h0
[15:13]	Reserved	Reserved	RO	-
[12]	cxl_cache_txn_timeout_enable	-	RW	1'h0
[11:8]	cxl_cache_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.cache.	RW	4'h0
[7:5]	Reserved	Reserved	RO	-
[4]	cxl_mem_txn_timeout_enable	When Set this bit enables CXL.mem Transaction Timeout mechanism.	RW	1'h0
[3:0]	cxl_mem_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.mem.	RW	4'h0

4.3.4.56 por_ccla_root_port_n_security_policy

Contains Root_Port_n_Security_Policy_Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF28`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-114: por_ccla_root_port_n_security_policy

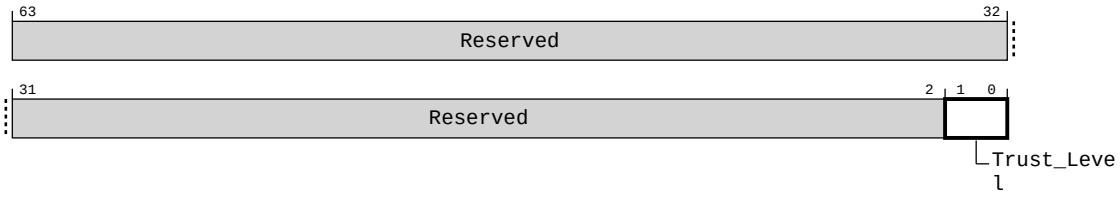


Table 4-130: por_ccla_root_port_n_security_policy attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	Trust_Level	Trust Level for the CXL.cache Device below Root Port n	RW	2'h2

4.3.4.57 por_ccla_root_port_n_id

Contains Root_Port_n_ID_Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF30`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-115: por_ccla_root_port_n_id

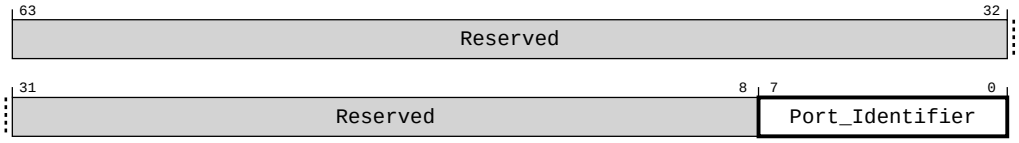


Table 4-131: por_ccla_root_port_n_id attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:0]	Port_Identifier	Port Identifier of the Root Port n	RW	8'h0

4.3.4.58 por_ccla_cxl_link_layer_defeature

CXL Link Layer Defeature Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE18`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxllink_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-116: por_ccla_cxl_link_layer_defeature

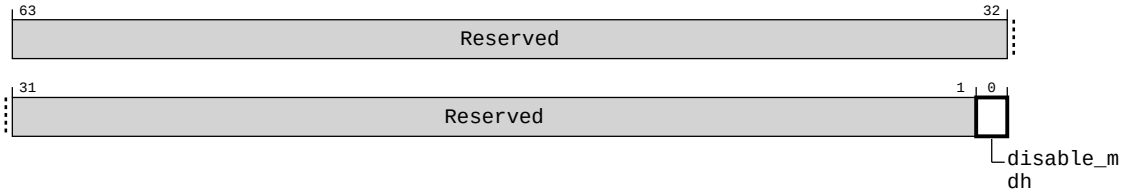


Table 4-132: por_ccla_cxl_link_layer_defeature attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	disable_mdh	Write 1 to disable MDH. Software needs to ensure it programs this value consistently on the UP & DP. After programming, a warm reset is required for the disable to take effect.	RW	1'b0

4.3.4.59 por_ccla_ull_ctl

Upper Link Layer Control Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE20`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ull_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-117: por_ccla_ull_ctl

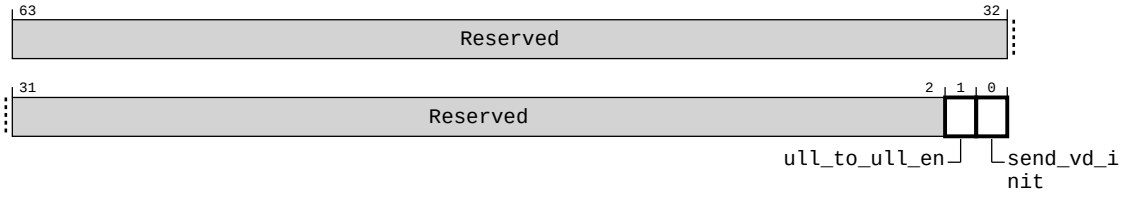


Table 4-133: por_ccla_ull_ctl attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	ull_to_ull_en	Used to enable ULL-to_ULL mode. Must be set on both sides of the link before 'send_vd_init' is set on either side **1'b0** When clear, ull-to-ull mode is disabled. **1'b1** When set, ull-to-ull mode is enabled.	RW	1'b0
[0]	send_vd_init	Used to send VD Init message. Must only be used for direct ULL to ULL connection. Must be used along with Tx ULL state (tx_ull_state) status bit **1'b0** When clear and tx_ull_state is in run_state, sends VD.De-activate flit. **1'b1** When set and tx_ull_state is in stop state, sends VD.Activate flit.	RW	1'b0

4.3.4.60 por_ccla_ull_status

Upper Link Layer Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE28`

Type

RO

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ull_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-118: por_ccla_ull_status

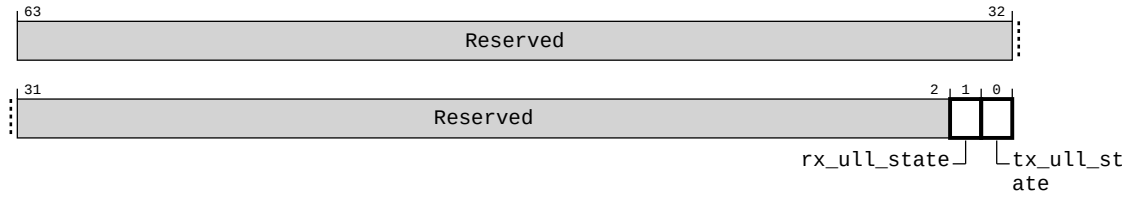


Table 4-134: por_ccla_ull_status attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	rx_ull_state	Reflects the Rx ULL state . **1'b0** Rx ULL is in Stop state **1'b1** Rx ULL is in Run state	RO	1'b0
[0]	tx_ull_state	Reflects the Tx ULL state . **1'b0** Tx ULL is in Stop state **1'b1** Tx ULL is in Run state	RO	1'b0

4.3.4.61 por_ccla_cxl_ll_errinject_ctl

CXL .cache .mem Link Layer Error Injection Control Register. Not used in Host mode

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE30`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxlerrinj_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-119: por_ccla_cxl_ll_errinject_ctl

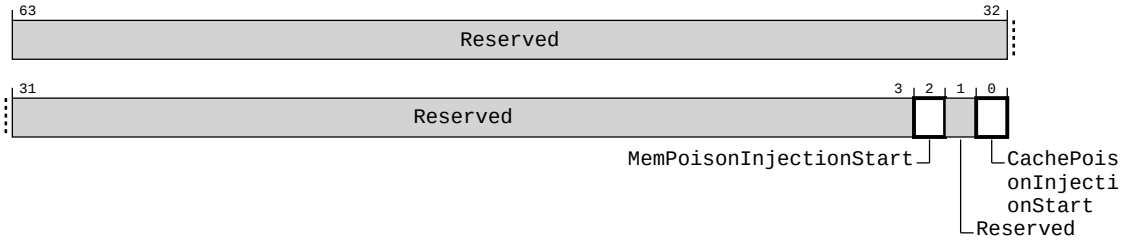


Table 4-135: por_ccla_cxl_ll_errinject_ctl attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	MemPoisonInjectionStart	Software writes 0x1 to this bit to trigger a single poison injection on a CXL.mem message in the Tx direction. Hardware must override the poison field in the data header slot ofthe corresponding message (DRS if device, RwD if Host). This bit is required only if CXL.mem protocol issupported.	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	CachePoisonInjectionStart	Software writes 0x1 to this bit to trigger a single poison injection on a CXL.cache message in the Tx direction. Hardware must override the poison field in the data header slot ofthe corresponding message (D2H if device, H2D if Host). This bit is required only if CXL.cache protocol issupported.	RW	1'b0

4.3.4.62 por_ccla_cxl_ll_errinject_stat

CXL .cache .mem Link Layer Error Injection Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE38`

Type

RO

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.cxlerrinj_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-120: por_ccla_cxl_ll_errinject_stat

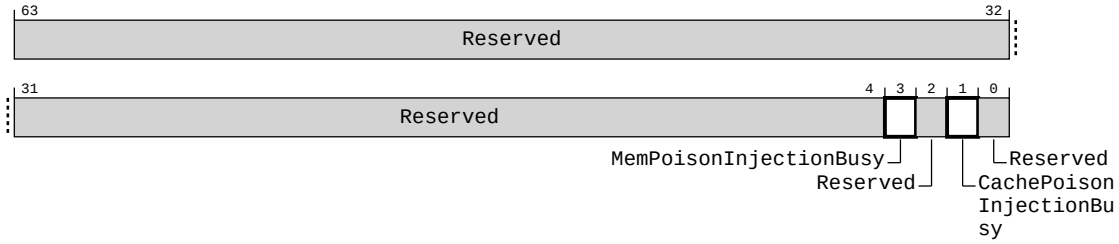


Table 4-136: por_ccla_cxl_ll_errinject_stat attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	MemPoisonInjectionBusy	Hardware loads 1 to this bit when the Start bit is written. Hardware must clear this bit to indicate that it has indeed finished poisoning a packet. Software is permitted to poll on this bit to find out when hardware has finished poison injection. This bit is required only if CXL.mem protocol issupported.	RO	1'b0
[2]	Reserved	Reserved	RO	-
[1]	CachePoisonInjectionBusy	Hardware loads 1 to this bit when the Start bit is written. Hardware must clear this bit to indicate that it has indeed finished poisoning a packet. Software is permitted to poll on this bit to find out when hardware has finished poison injection. This bit is required only if CXL.cache protocol issupported.	RO	1'b0
[0]	Reserved	Reserved	RO	-

4.3.4.63 por_ccla_cxl_viral_prop_en

Bit Vector which controls viral propagation. Each bit represents the logical ID of corresponding CML gateway block.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE40`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-121: por_ccla_cxl_viral_prop_en

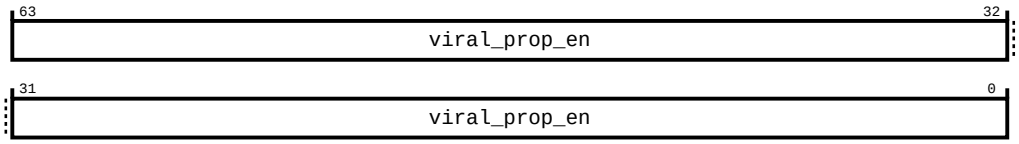


Table 4-137: por_ccla_cxl_viral_prop_en attributes

Bits	Name	Description	Type	Reset
[63:0]	viral_prop_en	Bit vector, where each bit represents logical ID of a CML gateway block present on CMN. Each bit when set, enables propagation of CXL Viral to that gateway block. **1'b0** Viral propagation is disabled **1'b1** Viral propagation is enabled	RW	64'b0

4.3.4.64 por_ccla_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2008

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-122: por_ccla_pmu_event_sel

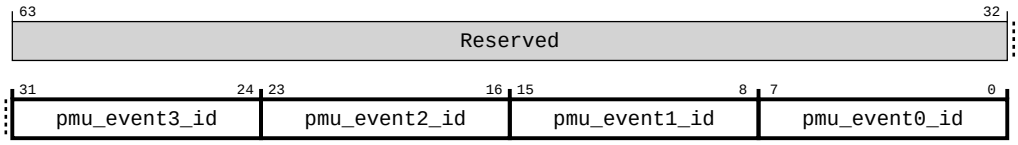


Table 4-138: por_ccla_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	pmu_event3_id	CCLA PMU Event 3 ID; see pmu_event0_id for encodings	RW	8'b0
[23:16]	pmu_event2_id	CCLA PMU Event 2 ID; see pmu_event0_id for encodings	RW	8'b0
[15:8]	pmu_event1_id	CCLA PMU Event 1 ID; see pmu_event0_id for encodings	RW	8'b0
[7:0]	pmu_event0_id	CCLA PMU Event 0 ID **8'h00** No event **8'h21** LA_RX_CXS : number of RX CXS beats **8'h22** LA_TX_CXS : number of TX CXS beats **8'h23** LA_RX_CXS_AVG_SIZE : average size of RX CXS beats **8'h24** LA_TX_CXS_AVG_SIZE : average size of TX CXS beats **8'h25** LA_TX_CXS_LCRD_BACKPRESSURE : CXS backpressue due to lack of CXS credits **8'h26** LA_LINK_CRDBUF_OCC : CCLA RX RAM buffer occupany **8'h27** LA_LINK_CRDBUF_ALLOC: CCLA RX RAM buffer allocation **8'h28** PFWD RCVR CXS beats **8'h29** PFWD SNDR NUM FLITS **8'h2A** PFWD SNDR Number of message stalls due to static credits **8'h2B** PFWD SNDR Number of message stalls due to dynamic credits	RW	8'b0

4.3.4.65 por_ccla_errfr

Functions as the error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3000`

Type

RO

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-123: por_ccla_errfr

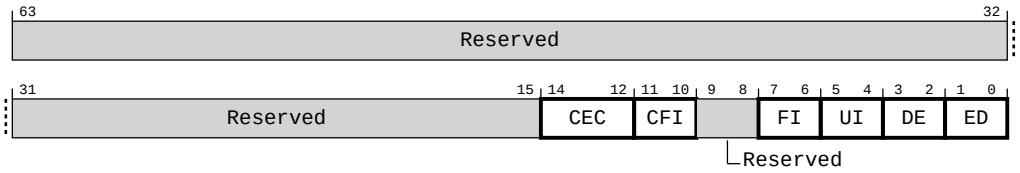


Table 4-139: por_ccla_errfr attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism **3'b000** Does not implement standardized error counter model **3'b010** Implements 8-bit error counter in por_ccla_errmisc[39:32] **3'b100** Implements 16-bit error counter in por_ccla_errmisc[47:32]	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10

Bits	Name	Description	Type	Reset
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors for data poison	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.4.66 por_ccla_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3008`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-124: por_ccla_errctlr

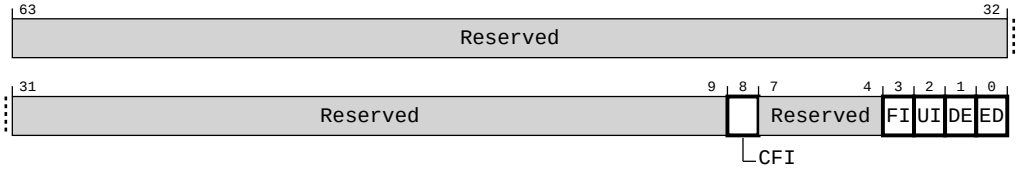


Table 4-140: por_ccla_errctlr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[8]	CFI	Enables corrected error interrupt as specified in por_ccla_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_ccla_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_ccla_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_ccla_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_ccla_errfr.ED	RW	1'b0

4.3.4.67 por_ccla_errstatus

Functions as the error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3010`

Type

W1C

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-125: por_ccla_errstatus

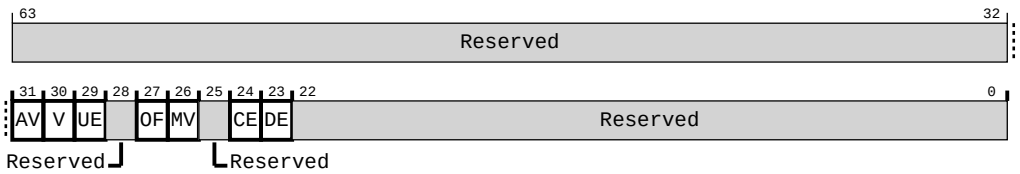


Table 4-141: por_ccla_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; por_ccla_erraddr contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	por_ccla_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0

Bits	Name	Description	Type	Reset
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.4.68 por_ccla_erraddr

Contains the error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3018

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-126: por_ccla_erraddr

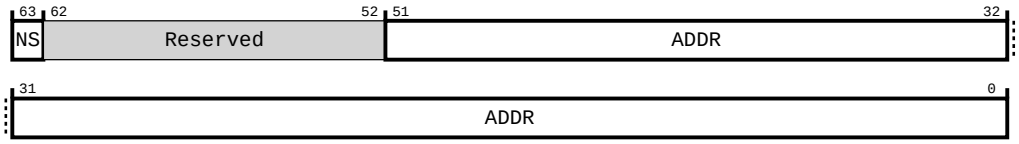


Table 4-142: por_ccla_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: por_ccla_erraddr.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.4.69 por_ccla_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3020`

Type

RW

Reset value

See individual bit resets

Secure group override

por_ccla_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-127: por_ccla_errmisc

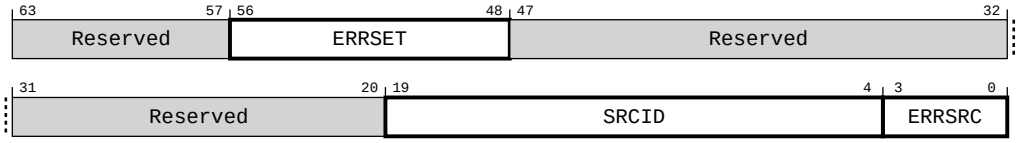


Table 4-143: por_ccla_errmisc attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ERRSET	RAM entry set address for parity error	RW	9'b0
[47:20]	Reserved	Reserved	RO	-
[19:4]	SRCID	CCIX RAID of the requestor or the snoop target	RW	16'b0
[3:0]	ERRSRC	Source of the parity error **4'b0000** Read data buffer 0 **4'b0001** Read data buffer 1 **4'b0010** Write data buffer 0 **4'b0011** Write data buffer 1 **4'b0100** Passive Buffer **4'b0101** CCLA Data RAM **4'b0110:** PortFwd Data RAM **4'b0111** CXL Viral **4'b1000** CXL.Mem Poison **4'b1001** CXL.Cache Poison	RW	4'b000

4.3.4.70 por_ccla_errfr_NS

Functions as the Non-secure error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3100`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-128: por_ccla_errfr_NS

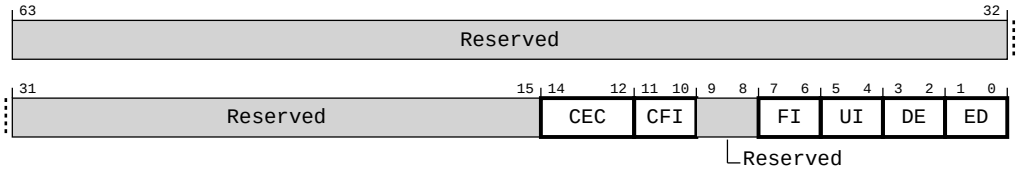


Table 4-144: por_ccla_errfr_NS attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism **3'b000** Does not implement standardized error counter model **3'b010** Implements 8-bit error counter in por_ccla_errmisc[39:32] **3'b100** Implements 16-bit error counter in por_ccla_errmisc[47:32]	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors for data poison	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.4.71 por_ccla_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3108

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-129: por_ccla_errctlr_NS

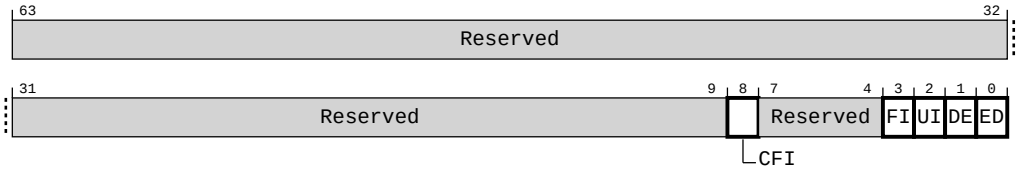


Table 4-145: por_ccla_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_ccla_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_ccla_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_ccla_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_ccla_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_ccla_errfr.ED	RW	1'b0

4.3.4.72 por_ccla_errstatus_NS

Functions as the Non-secure error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3110`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-130: por_ccla_errstatus_NS

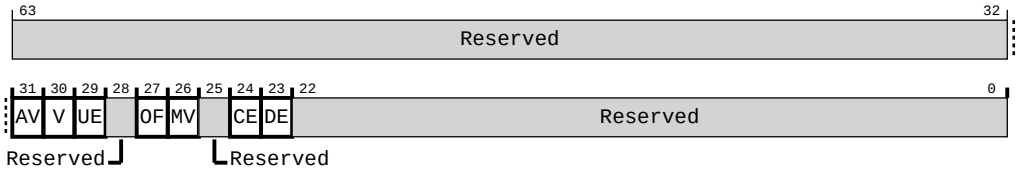


Table 4-146: por_ccla_errstatus_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; por_ccla_erraddr contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	por_ccla_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.4.73 por_ccla_erraddr_NS

Contains the Non-secure error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3118`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-131: por_ccla_erraddr_NS

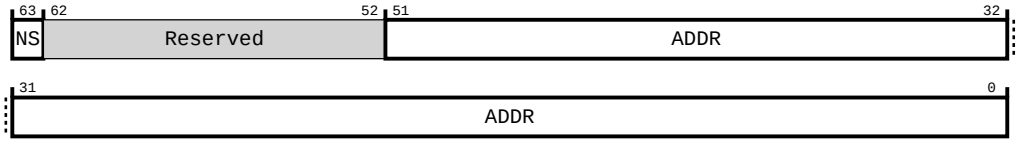


Table 4-147: por_ccla_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: por_ccla_erraddr.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.4.74 por_ccla_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3120`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-132: por_ccla_errmisc_NS

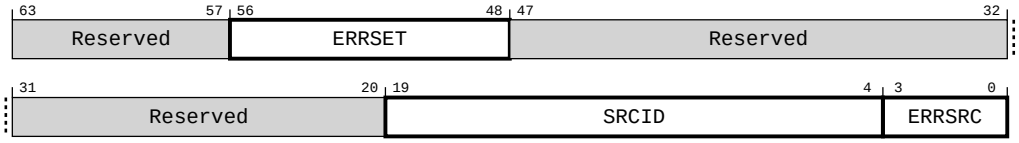


Table 4-148: por_ccla_errmisc_NS attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ERRSET	RAM entry set address for parity error	RW	9'b0
[47:20]	Reserved	Reserved	RO	-
[19:4]	SRCID	CCIX RAID of the requestor or the snoop target	RW	16'b0
[3:0]	ERRSRC	Source of the parity error **4'b0000** Read data buffer 0 **4'b0001** Read data buffer 1 **4'b0010** Write data buffer 0 **4'b0011** Write data buffer 1 **4'b0100** Passive Buffer **4'b0101** CCLA Data RAM **4'b0110:** PortFwd Data RAM **4'b0111** CXL Viral **4'b1000** CXL.Mem Poison **4'b1001** CXL.Cache Poison	RW	4'b000

4.3.5 Configuration manager register descriptions

This section lists the configuration registers.

4.3.5.1 por_cfgm_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-133: por_cfgm_node_info

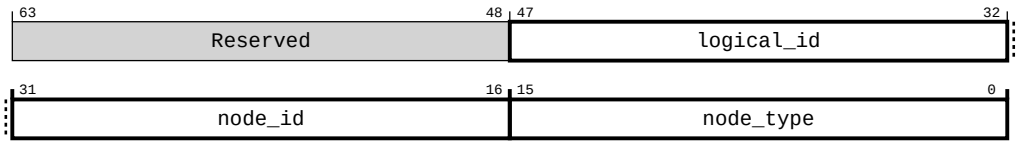


Table 4-149: por_cfgm_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0002

4.3.5.2 por_cfgm_periph_id_0_periph_id_1

Functions as the peripheral ID 0 and peripheral ID 1 register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h8

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-134: por_cfgm_periph_id_0_periph_id_1

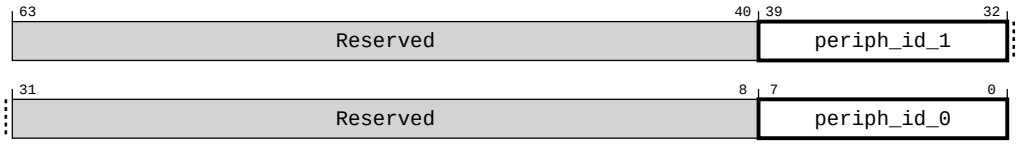


Table 4-150: por_cfgm_periph_id_0_periph_id_1 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_1	Peripheral ID 1	RO	8'b10110100
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_0	Peripheral ID 0	RO	Configuration dependent

4.3.5.3 por_cfgm_periph_id_2_periph_id_3

Functions as the peripheral ID 2 and peripheral ID 3 register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h10

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-135: por_cfgm_periph_id_2_periph_id_3

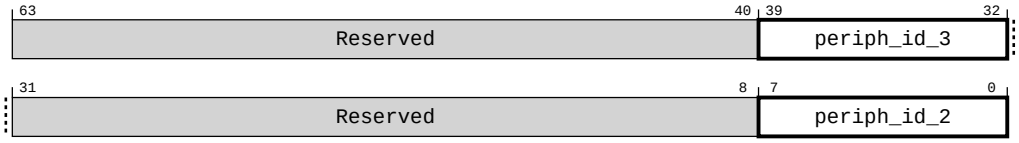


Table 4-151: por_cfgm_periph_id_2_periph_id_3 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_3	Peripheral ID 3	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_2	Peripheral ID 2 [7:4] indicates revision: 0x0 r0p0 0x1 r1p0 0x2 r2p0 0x3 r3p0 [3] JEDEC JEP106 identity code, 1'b1 [2:0] JEP106 identity code [6:4], 0'b011	RO	Configuration dependent

4.3.5.4 por_cfgm_periph_id_4_periph_id_5

Functions as the peripheral ID 4 and peripheral ID 5 register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h18`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-136: por_cfgm_periph_id_4_periph_id_5

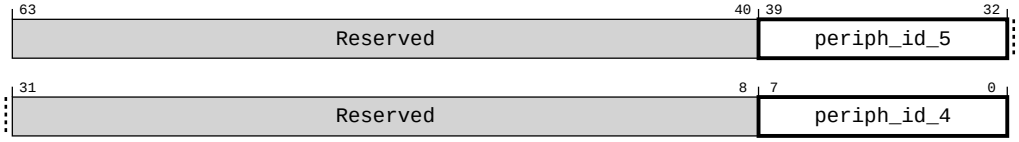


Table 4-152: por_cfgm_periph_id_4_periph_id_5 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_5	Peripheral ID 5	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_4	Peripheral ID 4	RO	8'b11000100

4.3.5.5 por_cfgm_periph_id_6_periph_id_7

Functions as the peripheral ID 6 and peripheral ID 7 register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h20`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-137: por_cfgm_periph_id_6_periph_id_7

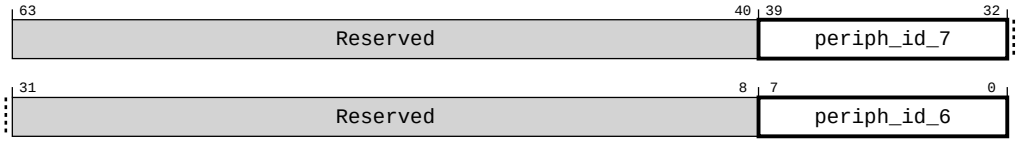


Table 4-153: por_cfgm_periph_id_6_periph_id_7 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	periph_id_7	Peripheral ID 7	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	periph_id_6	Peripheral ID 6	RO	8'b0

4.3.5.6 por_cfgm_component_id_0_component_id_1

Functions as the component ID 0 and component ID 1 register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h28`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-138: por_cfgm_component_id_0_component_id_1

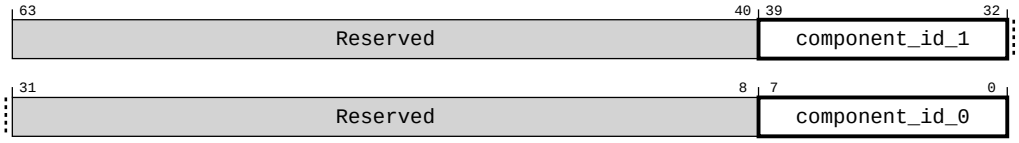


Table 4-154: por_cfgm_component_id_0_component_id_1 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	component_id_1	Component ID 1	RO	8'b11110000
[31:8]	Reserved	Reserved	RO	-
[7:0]	component_id_0	Component ID 0	RO	8'b00001101

4.3.5.7 por_cfgm_component_id_2_component_id_3

Functions as the component ID 2 and component ID 3 register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h30`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-139: por_cfgm_component_id_2_component_id_3

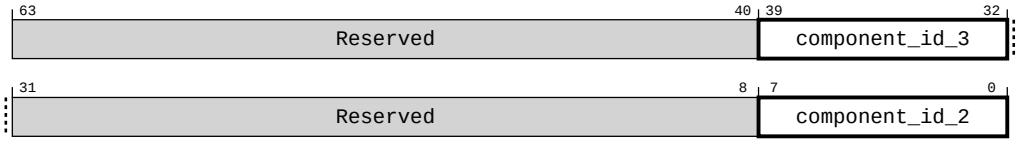


Table 4-155: por_cfgm_component_id_2_component_id_3 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	component_id_3	Component ID 3	RO	8'b10110001
[31:8]	Reserved	Reserved	RO	-
[7:0]	component_id_2	Component ID 2	RO	8'b00000101

4.3.5.8 por_cfgm_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-140: por_cfgm_child_info

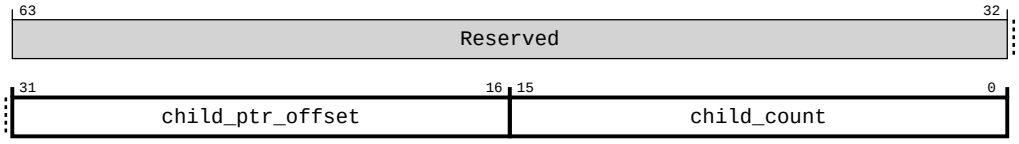


Table 4-156: por_cfgm_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h100
[15:0]	child_count	Number of child nodes; used in discovery process	RO	Configuration dependent

4.3.5.9 por_cfgm_secure_access

Functions as the Secure access control register. This register must be set up at boot time. Before initiating a write to this register, software must ensure that no other configuration accesses are in flight. Once this write is initiated, no other configuration accesses are initiated until complete.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-141: por_cfgm_secure_access

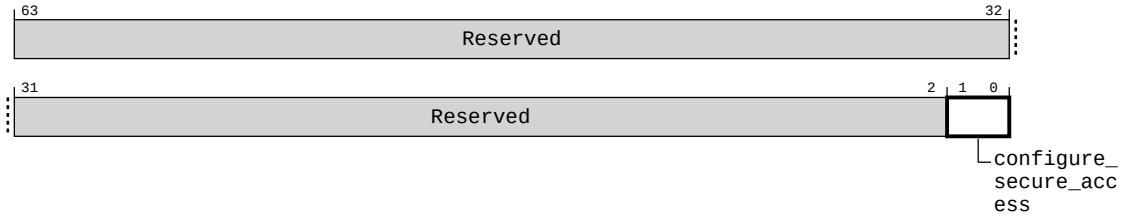


Table 4-157: por_cfgm_secure_access attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	configure_secure_access	Secure access mode 2'b00: Default operation 2'b01: Allows Non-secure access to Secure registers 2'b10: Allows Secure access only to any configuration register regardless of its security status 2'b11: Undefined behavior	RW	2'b0

4.3.5.10 por_cfgm_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h988`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-142: por_cfgm_secure_register_groups_override

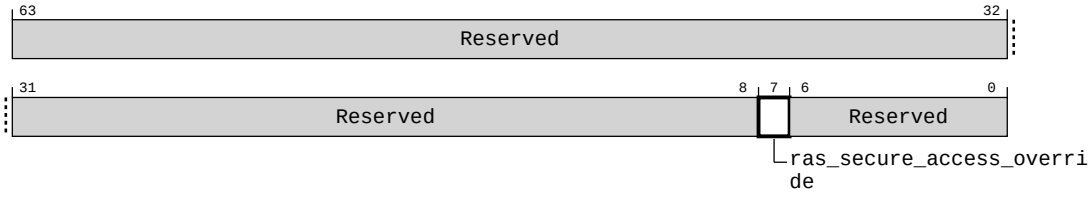


Table 4-158: por_cfgm_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7]	ras_secure_access_override	Allow Non-secure access to Secure RAS registers	RW	1'b0
[6:0]	Reserved	Reserved	RO	-

4.3.5.11 por_cfgm_errgsr_mxp_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the XP <n> Secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3000 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_cfgm_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-143: por_cfgm_errgsr_mxp_0-7

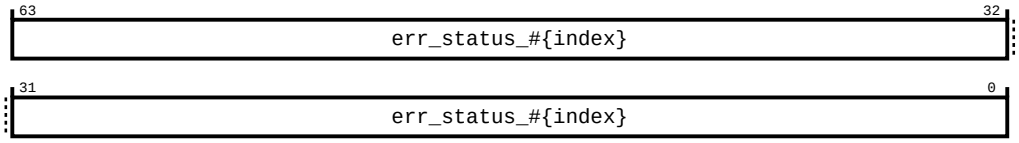


Table 4-159: por_cfgm_errgsr_mxp_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of MXP [Error/Fault] <n> status	RO	64'h0

4.3.5.12 por_cfgm_errgsr_mxp_0-7_NS

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the XP <n> Non-secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3040 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-144: por_cfgm_errgsr_mxp_0-7_NS

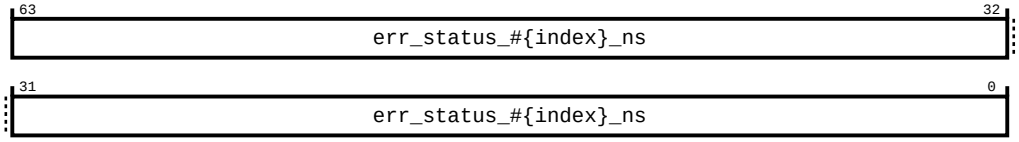


Table 4-160: por_cfgm_errgsr_mxp_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of MXP [Error/Fault] <n> status	RO	64'h0

4.3.5.13 por_cfgm_errgsr_hni_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the HNI <n> Secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3080 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_cfgm_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-145: por_cfgm_errgsr_hni_0-7

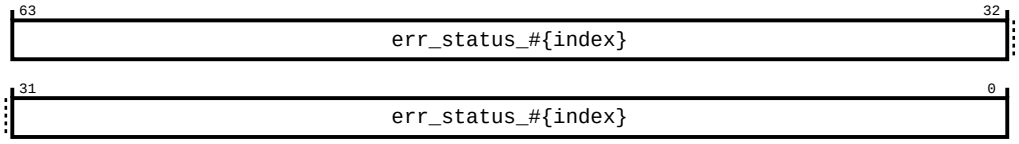


Table 4-161: por_cfgm_errgsr_hni_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of HNI [Error/Fault] <n> status	RO	64'h0

4.3.5.14 por_cfgm_errgsr_hni_0-7_NS

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the HNI <n> Non-secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h30C0 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-146: por_cfgm_errgsr_hni_0-7_NS

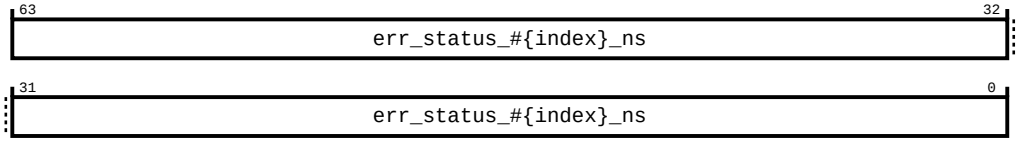


Table 4-162: por_cfgm_errgsr_hni_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of HNI [Error/Fault] <n> status	RO	64'h0

4.3.5.15 por_cfgm_errgsr_hnf_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the HNF <n> Secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3100 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_cfgm_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-147: por_cfgm_errgsr_hnf_0-7

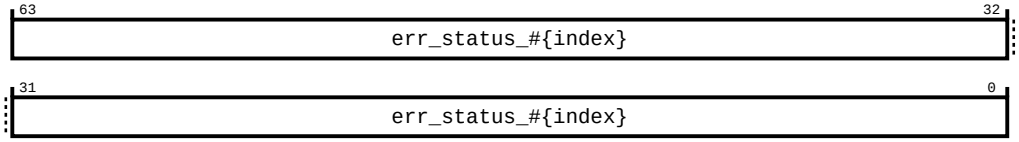


Table 4-163: por_cfgm_errgsr_hnf_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of HNF [Error/Fault] <n> status	RO	64'h0

4.3.5.16 por_cfgm_errgsr_hnf_0-7_NS

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the HNF <n> Non-secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3140 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-148: por_cfgm_errgsr_hnf_0-7_NS

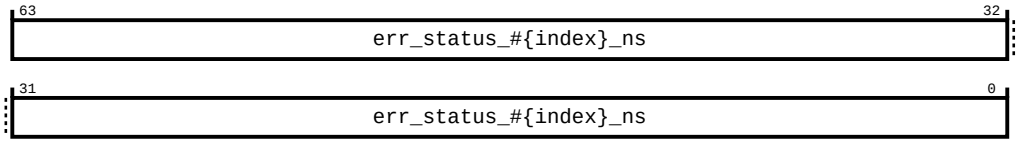


Table 4-164: por_cfgm_errgsr_hnf_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of HNF [Error/Fault] <n> status	RO	64'h0

4.3.5.17 por_cfgm_errgsr_sbsx_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the SBSX <n> Secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3180 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_cfgm_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-149: por_cfgm_errgsr_sbsx_0-7

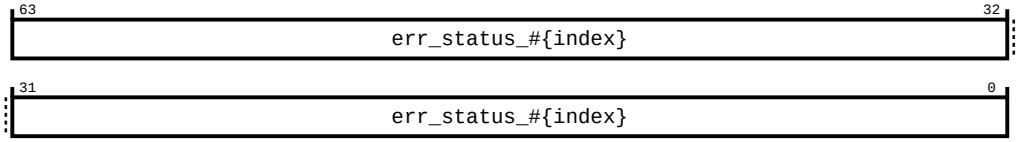


Table 4-165: por_cfgm_errgsr_sbsx_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of SBSX [Error/Fault] <n> status	RO	64'h0

4.3.5.18 por_cfgm_errgsr_sbsx_0-7_NS

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the SBSX <n> Non-secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h31C0 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-150: por_cfgm_errgsr_sbsx_0-7_NS

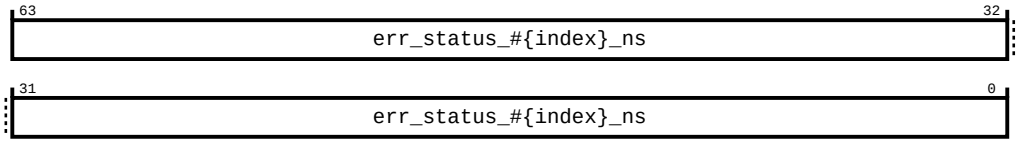


Table 4-166: por_cfgm_errgsr_sbsx_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of SBSX [Error/Fault] <n> status	RO	64'h0

4.3.5.19 por_cfgm_errgsr_cxg_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the CXG <n> Secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3200 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_cfgm_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-151: por_cfgm_errgsr_cxg_0-7

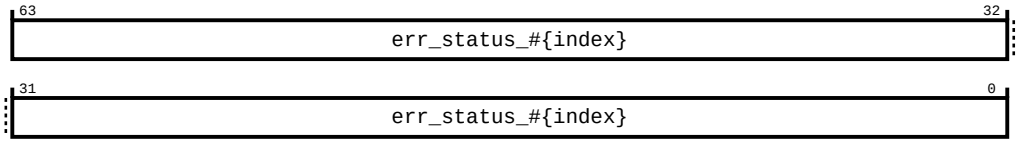


Table 4-167: por_cfgm_errgsr_cxg_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of CXG [Error/Fault] <n> status	RO	64'h0

4.3.5.20 por_cfgm_errgsr_cxg_0-7_NS

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the CXG <n> Non-secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3240 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-152: por_cfgm_errgsr_cxg_0-7_NS

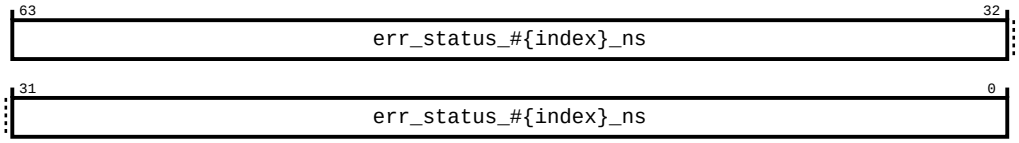


Table 4-168: por_cfgm_errgsr_cxg_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of CXG [Error/Fault] <n> status	RO	64'h0

4.3.5.21 por_cfgm_errgsr_mtsx_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the MTSX <n> Secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3280 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_cfgm_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-153: por_cfgm_errgsr_mtsx_0-7

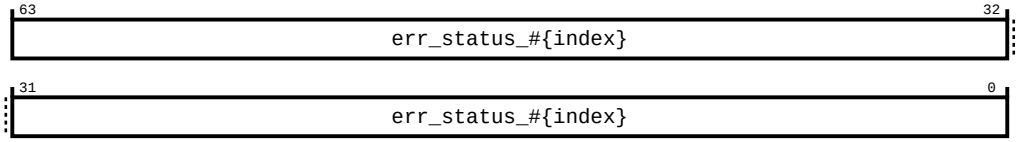


Table 4-169: por_cfgm_errgsr_mtsx_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of MTSX [Error/Fault] <n> status	RO	64'h0

4.3.5.22 por_cfgm_errgsr_mtsx_0-7_NS

There are 8 iterations of this register. The index ranges from 0 to 7. Provides the MTSX <n> Non-secure [Error/Fault] status. Where, Error occurs when the index is even and Fault occurs when the index is odd

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h32C0 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-154: por_cfgm_errgsr_mtsx_0-7_NS

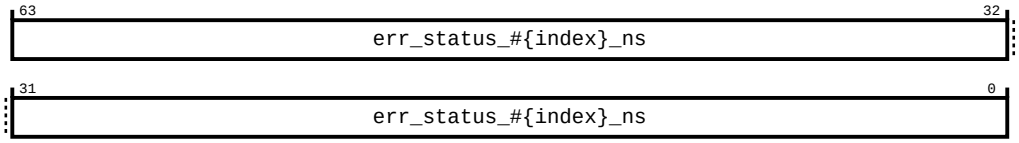


Table 4-170: por_cfgm_errgsr_mtsx_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of MTSX [Error/Fault] <n> status	RO	64'h0

4.3.5.23 por_cfgm_errdevaff

Functions as the device affinity register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FA8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-155: por_cfgm_errdevaff

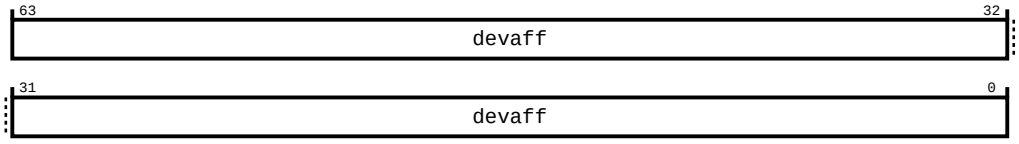


Table 4-171: por_cfgm_errdevaff attributes

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	64'b0

4.3.5.24 por_cfgm_errdevarch

Functions as the device architecture register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FB8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-156: por_cfgm_errdevarch

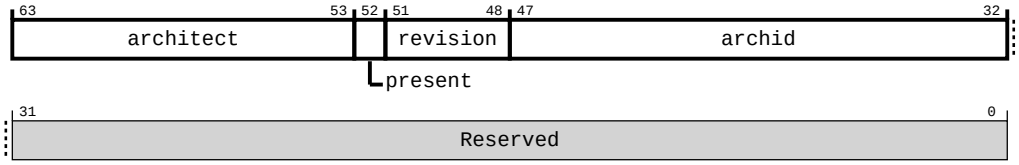


Table 4-172: por_cfgm_errdevarch attributes

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	11'h23B
[52]	present	Present	RO	1'b1
[51:48]	revision	Architecture revision	RO	4'b0
[47:32]	archid	Architecture ID	RO	16'h0A00
[31:0]	Reserved	Reserved	RO	-

4.3.5.25 por_cfgm_erridr

Contains the number of error records.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FC8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-157: por_cfgm_erridr

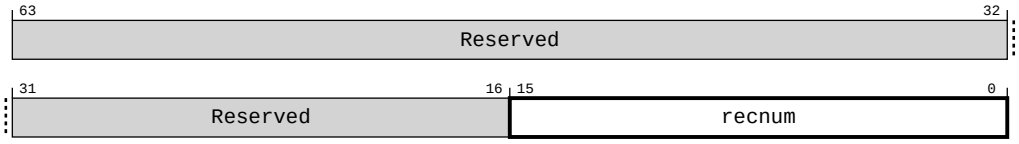


Table 4-173: por_cfgm_erridr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	recnum	Number of error records; equal to 2*(number of logical devices)	RO	Configuration dependent

4.3.5.26 por_cfgm_errpidr45

Functions as the identification register for peripheral ID 4 and peripheral ID 5.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FD0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-158: por_cfgm_errpidr45

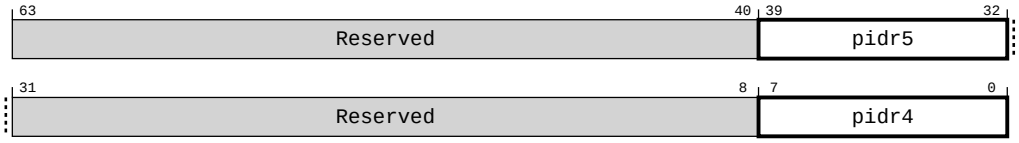


Table 4-174: por_cfgm_errpidr45 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr5	Peripheral ID 5	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr4	Peripheral ID 4	RO	8'h4

4.3.5.27 por_cfgm_errpidr67

Functions as the identification register for peripheral ID 6 and peripheral ID 7.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FD8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-159: por_cfgm_errpidr67

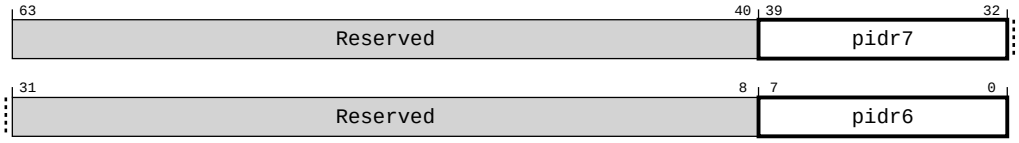


Table 4-175: por_cfgm_errpidr67 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr7	Peripheral ID 7	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr6	Peripheral ID 6	RO	8'b0

4.3.5.28 por_cfgm_errpidr01

Functions as the identification register for peripheral ID 0 and peripheral ID 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FE0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-160: por_cfgm_errpidr01

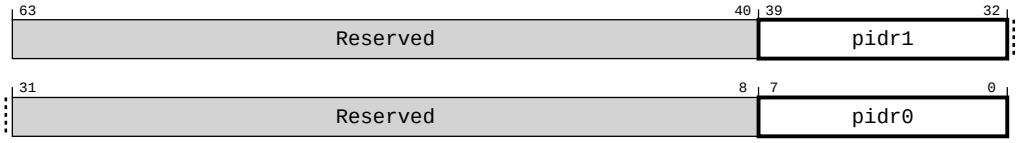


Table 4-176: por_cfgm_errpidr01 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr1	Peripheral ID 1	RO	8'hb4
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr0	Peripheral ID 0	RO	8'h34

4.3.5.29 por_cfgm_errpidr23

Functions as the identification register for peripheral ID 2 and peripheral ID 3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FE8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-161: por_cfgm_errpidr23

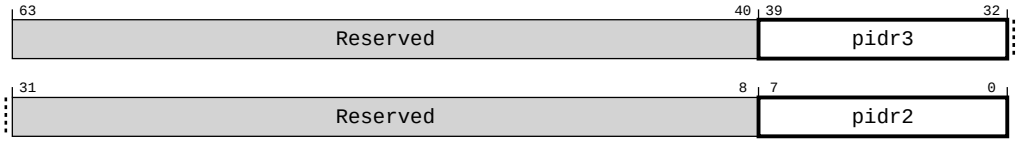


Table 4-177: por_cfgm_errpidr23 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr3	Peripheral ID 3	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr2	Peripheral ID 2	RO	8'h7

4.3.5.30 por_cfgm_errcidr01

Functions as the identification register for component ID 0 and component ID 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FF0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-162: por_cfgm_errcidr01

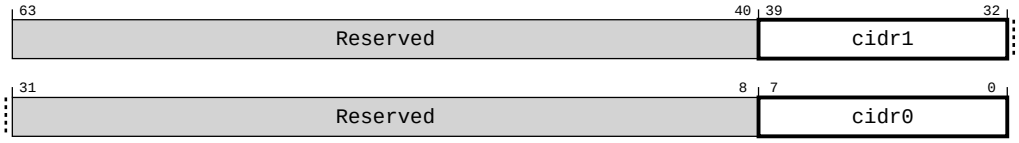


Table 4-178: por_cfgm_errcidr01 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	cidr1	Component ID 1	RO	8'hff
[31:8]	Reserved	Reserved	RO	-
[7:0]	cidr0	Component ID 0	RO	8'hd

4.3.5.31 por_cfgm_errcidr23

Functions as the identification register for component ID 2 and component ID 3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3FF8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-163: por_cfgm_errcidr23

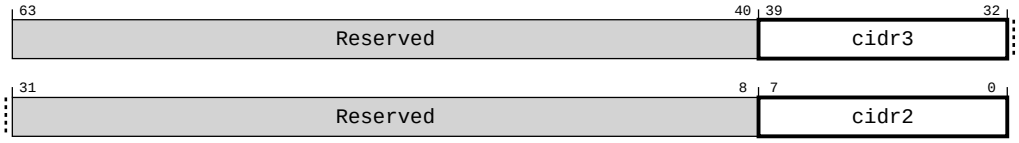


Table 4-179: por_cfgm_errcidr23 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	cidr3	Component ID 3	RO	8'hb1
[31:8]	Reserved	Reserved	RO	-
[7:0]	cidr2	Component ID 2	RO	8'h5

4.3.5.32 por_info_global

Contains user-specified values of build-time global configuration parameters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-164: por_info_global

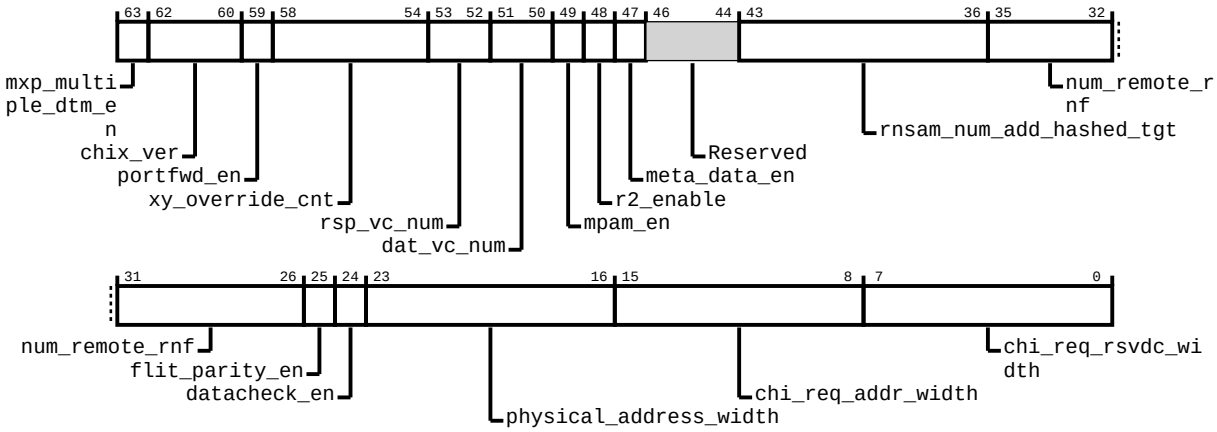


Table 4-180: por_info_global attributes

Bits	Name	Description	Type	Reset
[63]	mxp_multiple_dtm_en	Multiple DTMs feature enable. This is used if number of device ports on the XP is > 2	RO	Configuration dependent
[62:60]	chix_ver	CHIX Version Parameter: 2 -> CHIB, 3 -> CHIC, 4 -> CHID, 5 -> CHIE	RO	Configuration dependent
[59]	portfwd_en	CCIX Port to Port Forwarding feature enable	RO	Configuration dependent
[58:54]	xy_override_cnt	Number of Src-Tgt pairs whose XY route path can be overriden	RO	Configuration dependent
[53:52]	rsp_vc_num	Number of additional RSP channels internal to MXP. For inreased bandwidth, this parameter need to be set to 2	RO	Configuration dependent
[51:50]	dat_vc_num	Number of additional DAT channels internal to MXP. For inreased bandwidth, this parameter need to be set to 2	RO	Configuration dependent
[49]	mpam_en	MPAM enable	RO	Configuration dependent
[48]	r2_enable	CMN R2 feature enable	RO	Configuration dependent
[47]	meta_data_en	Meta Data Preservation mode enable	RO	Configuration dependent
[46:44]	Reserved	Reserved	RO	-
[43:36]	rnsam_num_add_hashed_tgt	Number of additional hashed target ID's supported by the RN SAM, beyond the local HNF count	RO	Configuration dependent
[35:26]	num_remote_rnf	Number of remote RN-F devices in the system when the CML feature is enabled	RO	Configuration dependent
[25]	flit_parity_en	Indicates whether parity checking is enabled in the transport layer on all flits sent on the interconnect	RO	Configuration dependent
[24]	datacheck_en	Indicates whether datacheck feature is enabled for CHI DAT flit	RO	Configuration dependent
[23:16]	physical_address_width	Physical address width	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[15:8]	chi_req_addr_width	REQ address width	RO	Configuration dependent
[7:0]	chi_req_rsvdc_width	RSVDC field width in CHI REQ flit	RO	Configuration dependent

4.3.5.33 por_info_global_1

Contains user-specified values of build-time global configuration parameters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-165: por_info_global_1

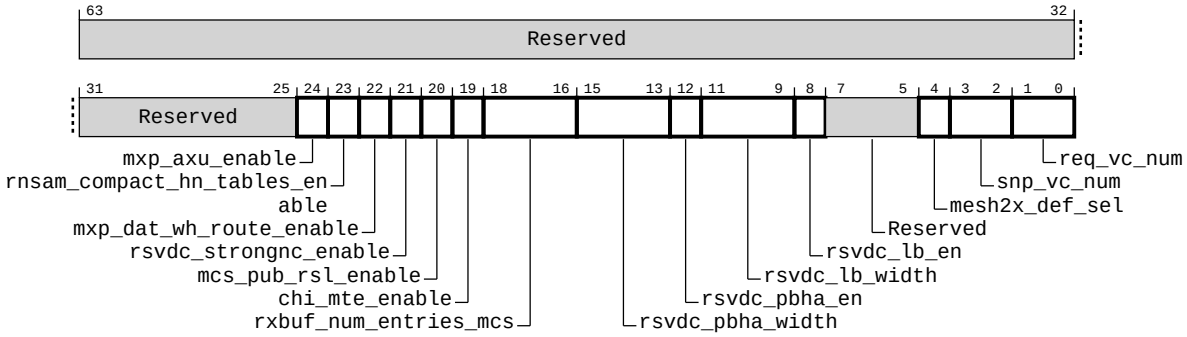


Table 4-181: por_info_global_1 attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	mxp_axu_enable	MXP AXU interface Enable	RO	Configuration dependent
[23]	rnsam_compact_hn_tables_enable	RNSAM Compact HN Tables Enable	RO	Configuration dependent
[22]	mxp_dat_wh_route_enable	Worm Hole Routing Enable for MXP DAT channel	RO	Configuration dependent
[21]	rsvdc_strongnc_enable	RSVDC StrongNC Mode Enable	RO	Configuration dependent
[20]	mcs_pub_rsl_enable	Register Slice enable for MCS PUB outputs	RO	Configuration dependent
[19]	chi_mte_enable	CHI MTE Feature Enable	RO	Configuration dependent
[18:16]	rxbuf_num_entries_mcs	RX Buffer Entries at upload interface of MCSX/MCSY	RO	Configuration dependent
[15:13]	rsvdc_pbha_width	RSVDC PBHA Field Width	RO	Configuration dependent
[12]	rsvdc_pbha_en	RSVDC PBHA Mode Enable	RO	Configuration dependent
[11:9]	rsvdc_lb_width	RSVDC Loop Back Field Width	RO	Configuration dependent
[8]	rsvdc_lb_en	RSVDC Loop Back Mode Enable	RO	Configuration dependent
[7:5]	Reserved	Reserved	RO	-
[4]	mesh2x_def_sel	Default ping-pong scheme selection for TGTID Lookup in 2xMESH	RO	Configuration dependent
[3:2]	snp_vc_num	Number of additional SNP channels internal to MXP. For inreased bandwidth, this parameter need to be set to 2	RO	Configuration dependent
[1:0]	req_vc_num	Number of additional REQ channels internal to MXP. For inreased bandwidth, this parameter need to be set to 2	RO	Configuration dependent

4.3.5.34 por_ppu_int_enable

Configures the HN-F PPU event interrupt. Contains the interrupt mask.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C00`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-166: por_ppu_int_enable

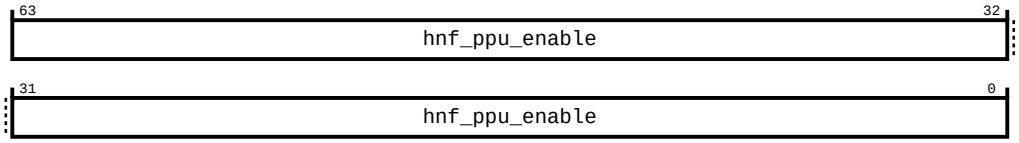


Table 4-182: por_ppu_int_enable attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_enable	Interrupt mask	RW	64'b0

4.3.5.35 por_ppu_int_enable_1

Configures the HN-F PPU event interrupt. Contains the interrupt mask.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-167: por_ppu_int_enable_1

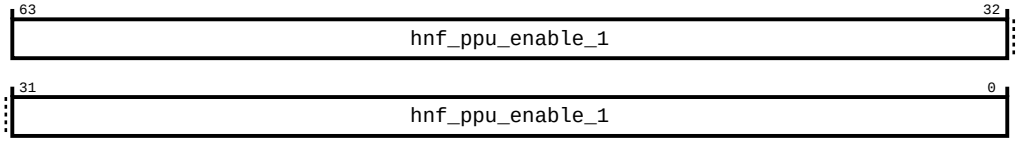


Table 4-183: por_ppu_int_enable_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_enable_1	Interrupt mask	RW	64'b0

4.3.5.36 por_ppu_int_status

Provides HN-F PPU event interrupt status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C10`

Type

W1C

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-168: por_ppu_int_status

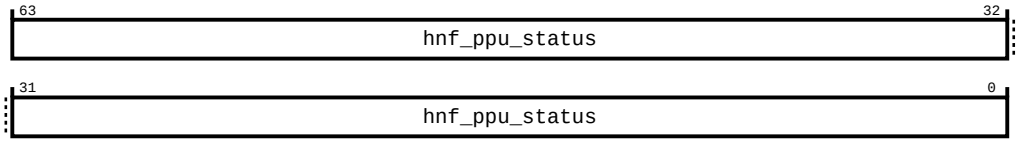


Table 4-184: por_ppu_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_status	Interrupt status	W1C	64'b0

4.3.5.37 por_ppu_int_status_1

Provides HN-F PPU event interrupt status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C18`

Type

W1C

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-169: por_ppu_int_status_1

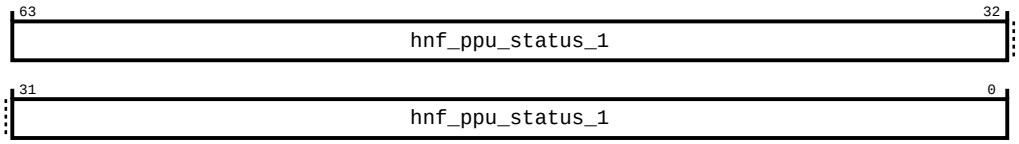


Table 4-185: por_ppu_int_status_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_status_1	Interrupt status	W1C	64'b0

4.3.5.38 por_ppu_qactive_hyst

Number of hysteresis clock cycles to retain QACTIVE assertion

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C20`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-170: por_ppu_qactive_hyst

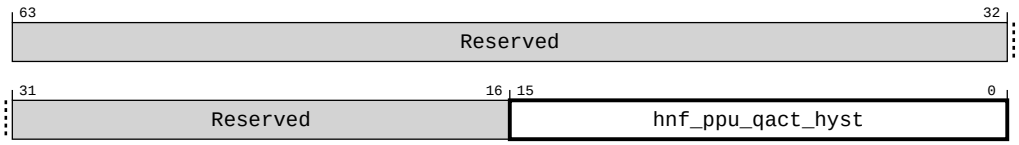


Table 4-186: por_ppu_qactive_hyst attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_ppu_qact_hyst	QACTIVE hysteresis	RW	16'h10

4.3.5.39 por_mpam_s_err_int_status

Provides HN-F MPAM Secure Error interrupt status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C28`

Type

W1C

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-171: por_mpam_s_err_int_status

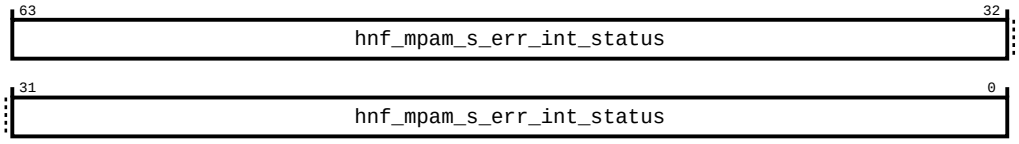


Table 4-187: por_mpam_s_err_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_s_err_int_status	MPAM S Interrupt status	W1C	64'b0

4.3.5.40 por_mpam_s_err_int_status_1

Provides HN-F MPAM Secure Error interrupt status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C30`

Type

W1C

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-172: por_mpam_s_err_int_status_1

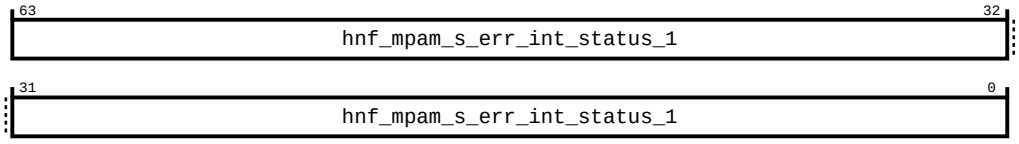


Table 4-188: por_mpam_s_err_int_status_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_s_err_int_status_1	MPAM S Interrupt status	W1C	64'b0

4.3.5.41 por_mpam_ns_err_int_status

Provides HN-F MPAM Non-secure Error interrupt status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C38`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-173: por_mpam_ns_err_int_status

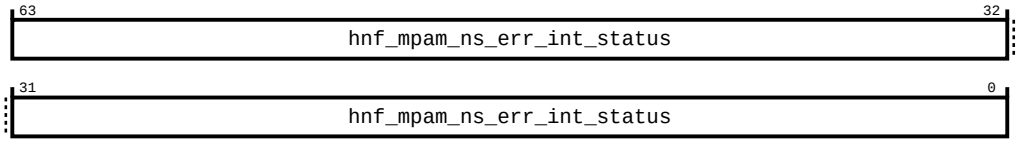


Table 4-189: por_mpam_ns_err_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_ns_err_int_status	MPAM NS Interrupt status	W1C	64'b0

4.3.5.42 por_mpam_ns_err_int_status_1

Provides HN-F MPAM Non-secure Error interrupt status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C40`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-174: por_mpam_ns_err_int_status_1

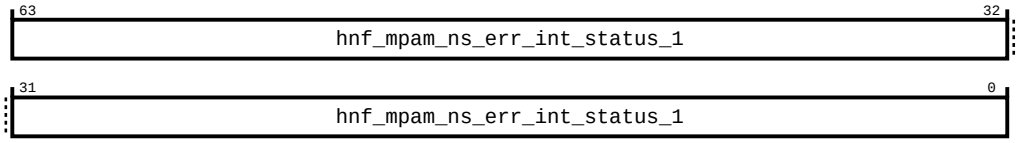


Table 4-190: por_mpam_ns_err_int_status_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_ns_err_int_status_1	MPAM NS Interrupt status	W1C	64'b0

4.3.5.43 por_cfgm_child_pointer_0-255

There are 256 iterations of this register. The index ranges from 0 to 255. Contains base address of child configuration node. NOTE: There will be as many child pointer registers in the Global Config Unit as the number of XP's on the chip. Each successive child pointer register will be at the next 8 byte address boundary. Each successive child pointer register will be named with the suffix corresponding to the register number. For example por_cfgm_child_pointer_<0:255>

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h100 + #{8 * index}
```

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-175: por_cfgm_child_pointer_0-255

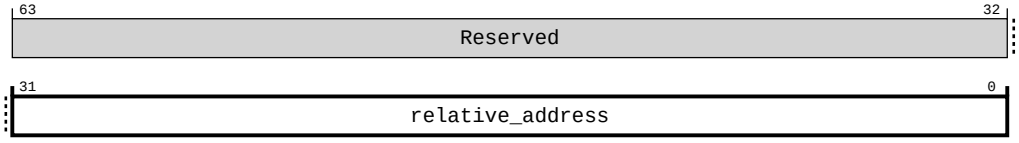


Table 4-191: por_cfgm_child_pointer_0-255 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	relative_address	Bits: **[31]** External or internal child node **[30]** Set to 1'b0 Bits **[29:0]** Child node address offset relative to PERIPHBASE **1'b1** Indicates child pointer points to a configuration node that is external to CMN-700 **1'b0** Indicates child pointer points to a configuration node that is internal to CMN-700	RO	32'b0

4.3.6 CXLAPB register descriptions

This section lists the CXLAPB registers.

4.3.6.1 por_cxlapb_link_rx_credit_ctl

CXL Link Rx Credit Control Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1110`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-176: por_cxlapb_link_rx_credit_ctl

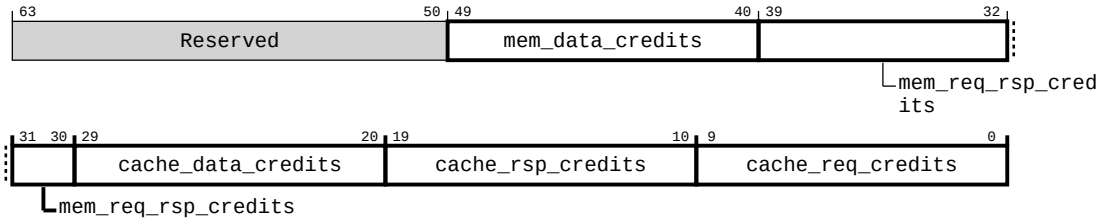


Table 4-192: por_cxlapb_link_rx_credit_ctl attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Credits to advertise for Mem Data channel at init	RW	Configuration dependent
[39:30]	mem_req_rsp_credits	Credits to advertise for Mem Request or Response channel at init	RW	Configuration dependent
[29:20]	cache_data_credits	Credits to advertise for Cache Data channel at init	RW	Configuration dependent
[19:10]	cache_rsp_credits	Credits to advertise for Cache Response channel at init	RW	Configuration dependent
[9:0]	cache_req_credits	Credits to advertise for Cache Request channel at init	RW	Configuration dependent

4.3.6.2 por_cxlapb_link_rx_credit_return_stat

CXL Link Rx Credit Return Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1118`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-177: por_cxlapb_link_rx_credit_return_stat

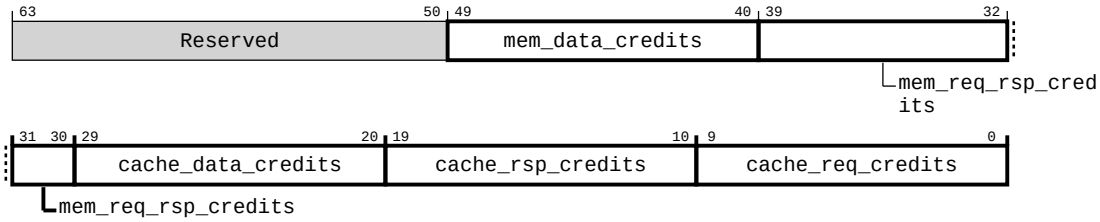


Table 4-193: por_cxlapb_link_rx_credit_return_stat attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits to be returned	RO	10'b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits to be returned	RO	10'b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits to be returned	RO	10'b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits to be returned	RO	10'b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits to be returned	RO	10'b0

4.3.6.3 por_cxlapb_link_tx_credit_stat

CXL Link Tx Credit Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1120`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-178: por_cxlapb_link_tx_credit_stat

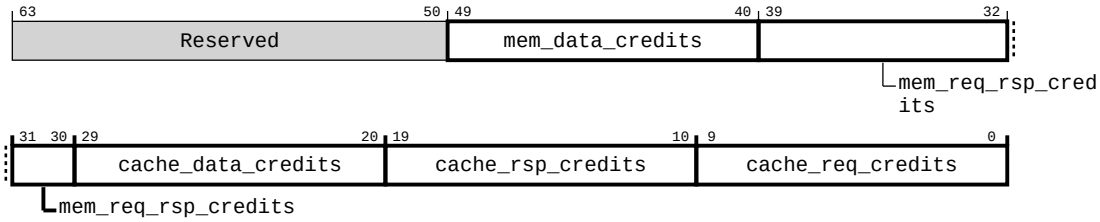


Table 4-194: por_cxlapb_link_tx_credit_stat attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	mem_data_credits	Running snapshot of accumulated Mem Data credits for TX	RO	10'b0
[39:30]	mem_req_rsp_credits	Running snapshot of accumulated Mem Request or Response credits for TX	RO	10'b0
[29:20]	cache_data_credits	Running snapshot of accumulated Cache Data credits for TX	RO	10'b0
[19:10]	cache_rsp_credits	Running snapshot of accumulated Cache Response credits for TX	RO	10'b0
[9:0]	cache_req_credits	Running snapshot of accumulated Cache Request credits for TX	RO	10'b0

4.3.6.4 por_cxlapb_cxl_security_policy

Contains CXL Security Policy

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1060`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-179: por_cxlapb_cxl_security_policy

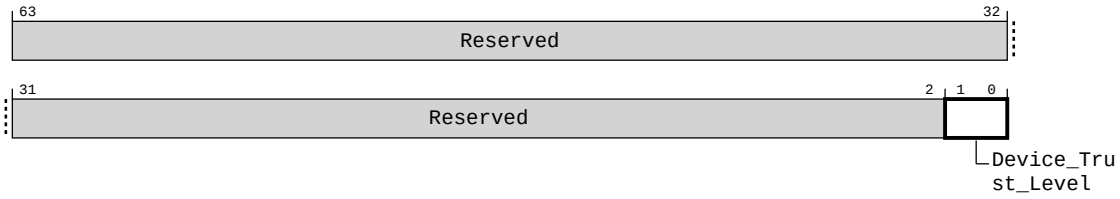


Table 4-195: por_cxlapb_cxl_security_policy attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	Device_Trust_Level	0 --> Trusted CXL Device. At this setting a CXL Device will be able to get access on CXL.cache for both host-attached and device attached memory ranges. The Host can still protect security sensitive memory regions. '1 --> Trusted for Device Attached Memory Range Only. At this setting a CXL Device will be able to get access on CXL.cache for device attached memory ranges only. Requests on CXL.cache for host-attached memory ranges will be aborted by the Host. '2 --> Untrusted CXL Device. At this setting all requests on CXL.cache will be aborted by the Host. Please note that these settings only apply to requests on CXL.cache. The device can still source requests on CXL.io regardless of these settings. Protection on CXL.io will be implemented using IOMMU based page tables. Default value of this field is 2.	RW	2'h2

4.3.6.5 por_cxlapb_cxl_hdm_decoder_capability

Contains CXL_HDM_Decoder_Capability_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1200`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-180: por_cxlapb_cxl_hdm_decoder_capability

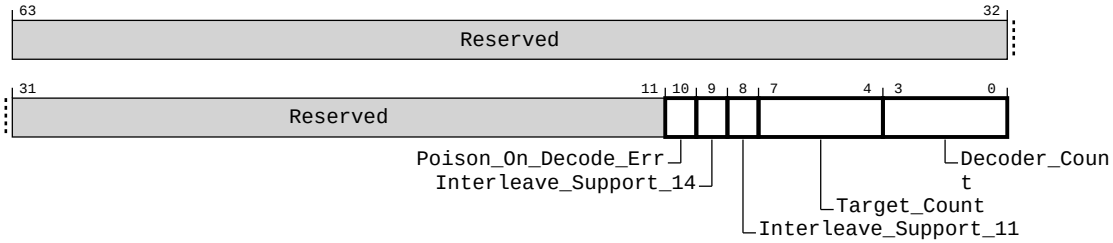


Table 4-196: por_cxlapb_cxl_hdm_decoder_capability attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	Poison_On_Decode_Err	If set the component is capable of returning poison on read access to addresses that are not positively decoded by any HDM Decoders in this component. If clear the component is not capable of returning poison under such scenarios.	RO	1'h0
[9]	Interleave_Support_14	If set the component supports interleaving based on Address bit 14 Address bit 13 and Address bit 12. Root ports and switches shall always set this bit indicating support for interleaving based on Address bits 14-12.	RO	1'h0
[8]	Interleave_Support_11	If set the component supports interleaving based on Address bit 11 Address bit 10 Address bit 9 and Address bit 8. Root Ports and Upstream Switch Ports shall always set this bit indicating support for interleaving based on Address bit 11-8.	RO	1'h0
[7:4]	Target_Count	The number of target ports each decoder supports (applicable to Upstream Switch Port and Root Port only). Maximum of 8. **1** 1 target port **2** 2 target ports **8** 8 target ports **Other** Reserved	RO	4'h1
[3:0]	Decoder_Count	Reports the number of memory address decoders implemented by the component. **0** 1 Decoder **1** 2 Decoders **2** 4 Decoders **3** 6 Decoders **4** 8 Decoders **5** 10 Decoders **Others** Reserved	RO	4'h0

4.3.6.6 por_cxlapb_cxl_hdm_decoder_global_control

Contains CXL_HDM_Decoder_Global_Control_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1204`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-181: por_cxlapb_cxl_hdm_decoder_global_control

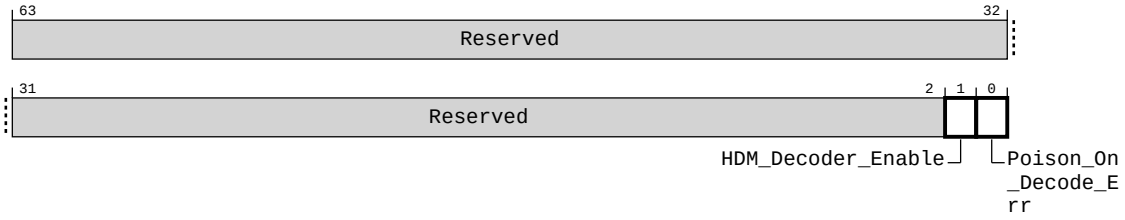


Table 4-197: por_cxlapb_cxl_hdm_decoder_global_control attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	HDM_Decoder_Enable	This bit is only applicable to CXL.mem devices and shall return 0 on Root Ports and Upstream Switch Ports. When this bit is set device shall use HDM decoders to decode CXL.mem transactions and not use HDM Base registers in DVSEC ID 0. Root Ports and Upstream Switch Ports always use HDM Decoders to decode CXL.mem transactions.	RW	1'h0
[0]	Poison_On_Decode_Err	This bit is RO and is hard-wired to 0 if Poison On Decode Error Capability=0. If set the component returns poison on read access to addresses that are not positively decoded by the component. If clear the component returns all 1s data without a poison under such scenarios.	RW	1'h0

4.3.6.7 por_cxlapb_cxl_hdm_decoder_0_base_low

Contains CXL_HDM_Decoder_0_Base_High_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1210`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-182: por_cxlapb_cxl_hdm_decoder_0_base_low

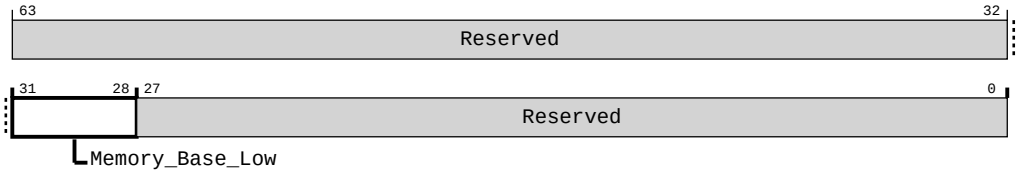


Table 4-198: por_cxlapb_cxl_hdm_decoder_0_base_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of the base of the address range managed by decoder 0	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.6.8 por_cxlapb_cxl_hdm_decoder_0_base_high

Contains CXL_HDM_Decoder_0_Base_Low_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1214`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-183: por_cxlapb_cxl_hdm_decoder_0_base_high

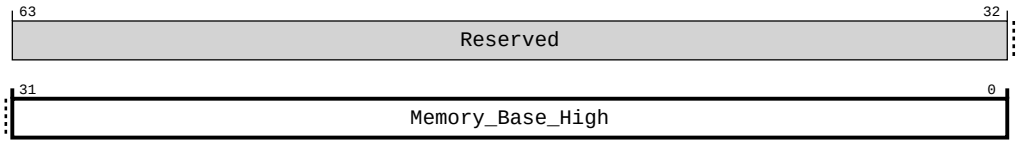


Table 4-199: por_cxlapb_cxl_hdm_decoder_0_base_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of the base of the address range managed by decoder 0.	RWL	32'h0

4.3.6.9 por_cxlapb_cxl_hdm_decoder_0_size_low

Contains CXL_HDM_Decoder_0_Size_Low_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1218`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-184: por_cxlapb_cxl_hdm_decoder_0_size_low

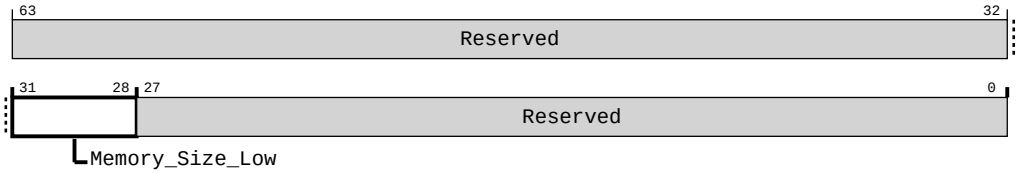


Table 4-200: por_cxlapb_cxl_hdm_decoder_0_size_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Size_Low	Corresponds to bits 31:28 of the size of the address range managed by decoder 0	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.6.10 por_cxlapb_cxl_hdm_decoder_0_size_high

Contains CXL_HDM_Decoder_0_Size_High_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h121C

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-185: por_cxlapb_cxl_hdm_decoder_0_size_high

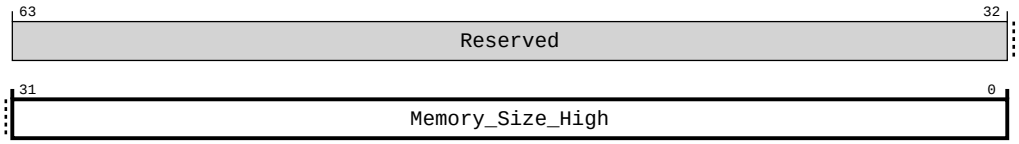


Table 4-201: por_cxlapb_cxl_hdm_decoder_0_size_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Size_High	Corresponds to bits 63:32 of the size of address range managed by decoder 0.	RWL	32'h0

4.3.6.11 por_cxlapb_cxl_hdm_decoder_0_control

Contains CXL_HDM_Decoder_0_Control_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1220

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-186: por_cxlapb_cxl_hdm_decoder_0_control

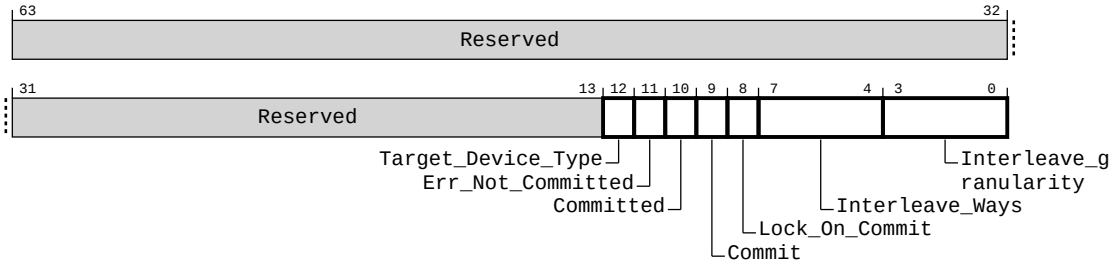


Table 4-202: por_cxlapb_cxl_hdm_decoder_0_control attributes

Bits	Name	Description	Type	Reset
[63:13]	Reserved	Reserved	RO	-
[12]	Target_Device_Type	**0** Target is a CXL Type 2 Device **1** Target is a CXL Type 3 Device	RWL	1'h0
[11]	Err_Not_Committed	Indicates the decode programming had an error and decoder is not active.	RWL	1'h0
[10]	Committed	Indicates the decoder is active	RWL	1'h0
[9]	Commit	Software sets this to 1 to commit this decoder	RWL	1'h0
[8]	Lock_On_Commit	If set all RWL fields in Decoder 0 registers will become read only when Committed changes to 1.	RWL	1'h0
[7:4]	Interleave_Ways	The number of targets across which this memory range is interleaved. 0 - 1 way 1 - 2 way 2 4 way 3 8 way All other reserved	RWL	4'h0
[3:0]	Interleave_granularity	The number of consecutive bytes that are assigned to each target in the Target List. **0** 256 Bytes **1** 512 Bytes **2** 1024 Bytes (1KB) **3** 2048 Bytes (2KB) **4** 4096 Bytes (4KB) **5** 8192 Bytes (8 KB) **6** 16384 Bytes (16 KB)	RWL	4'h0

4.3.6.12 por_cxlapb_cxl_hdm_decoder_0_dpa_skip_low

Contains CXL_HDM_Decoder_0_DPA_Skip_Low_Register. Only applicable for device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1224`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-187: por_cxlapb_cxl_hdm_decoder_0_dpa_skip_low

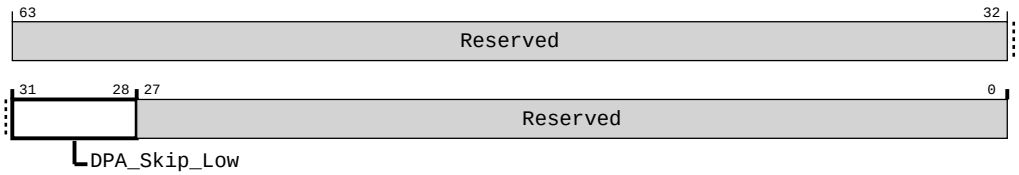


Table 4-203: por_cxlapb_cxl_hdm_decoder_0_dpa_skip_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	DPA_Skip_Low	Corresponds to bits 31:28 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.6.13 por_cxlapb_cxl_hdm_decoder_0_dpa_skip_high

Contains CXL_HDM_Decoder_0_DPA_Skip_High_Register. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1228

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-188: por_cxlapb_cxl_hdm_decoder_0_dpa_skip_high

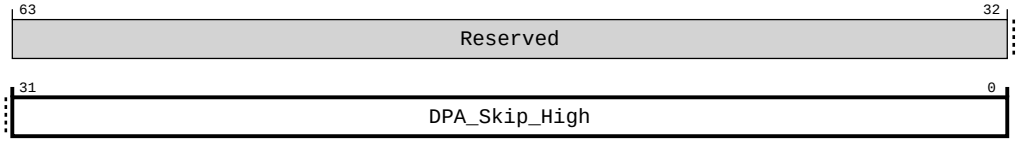


Table 4-204: por_cxlapb_cxl_hdm_decoder_0_dpa_skip_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	DPA_Skip_High	Corresponds to bits 63:32 of the DPA Skip length which when non-zero specifies a length of DPA space that is skipped unmapped by any decoder prior to the HPA to DPA mapping provided by this decoder.	RWL	32'h0

4.3.6.14 por_cxlapb_snoop_filter_group_id

Contains Snoop_Filter_Group_ID

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1800

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-189: por_cxlapb_snoop_filter_group_id

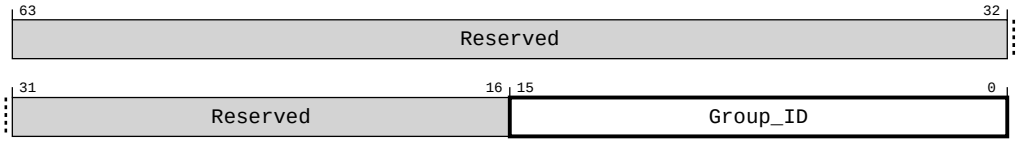


Table 4-205: por_cxlapb_snoop_filter_group_id attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	Group_ID	Uniquely identifies a snoop filter instance that is used to track CXL.cache devices below this Port. All Ports that share a single Snoop Filter instance shall set this field to the same value.	RO	16'h0

4.3.6.15 por_cxlapb_snoop_filter_effective_size

Contains Snoop_Filter_Effective_Size

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1804`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-190: por_cxlapb_snoop_filter_effective_size

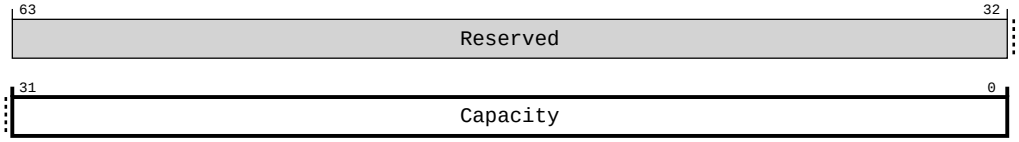


Table 4-206: por_cxlapb_snoop_filter_effective_size attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Capacity	Effective Snoop Filter Capacity representing the size of cache that can be effectively tracked by the Snoop Filter with this Group ID in multiples of 64K.	RO	32'h0

4.3.6.16 por_cxlapb_dvsec_cxl_range_1_base_high

Contains DVSEC_CXL_Range_1_Base_High. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h120`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-191: por_cxlapb_dvsec_cxl_range_1_base_high

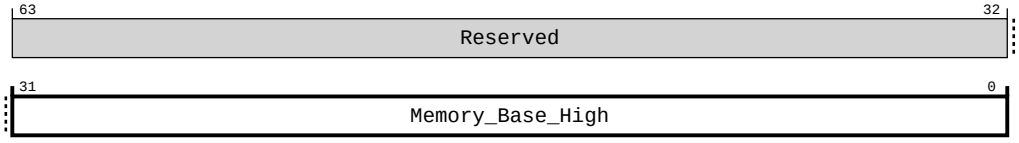


Table 4-207: por_cxlapb_dvsec_cxl_range_1_base_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder 0 Base High Register for backward compatibility reasons.	RWL	32'h0

4.3.6.17 por_cxlapb_dvsec_cxl_range_1_base_low

Contains DVSEC_CXL_Range_1_Base_Low. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h124`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-192: por_cxlapb_dvsec_cxl_range_1_base_low

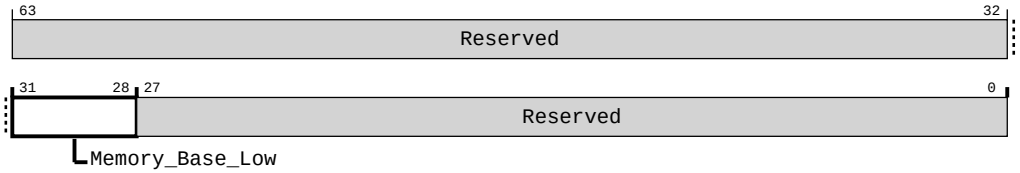


Table 4-208: por_cxlapb_dvsec_cxl_range_1_base_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 1 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match CXL HDM Decoder 0 Base Low Register for backward compatibility reasons.	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.6.18 por_cxlapb_dvsec_cxl_range_2_base_high

Contains DVSEC_CXL_Range_2_Base_High. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h130`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-193: por_cxlapb_dvsec_cxl_range_2_base_high

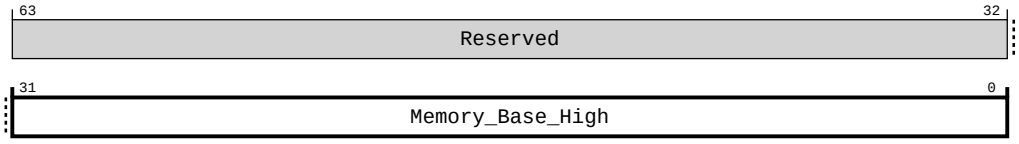


Table 4-209: por_cxlapb_dvsec_cxl_range_2_base_high attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Memory_Base_High	Corresponds to bits 63:32 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK. If a device implements CXL HDM Decoder Capability registers and software has enabled HDM Decoder by setting HDM Decoder Enable bit in CXL HDM Decoder Global Control register the value of this field is not used during address decode. It is recommended that software program this to match the corresponding CXL HDM Decoder Base High Register for backward compatibility reasons.	RWL	32'h0

4.3.6.19 por_cxlapb_dvsec_cxl_range_2_base_low

Contains DVSEC_CXL_Range_2_Base_Low. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h134`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-194: por_cxlapb_dvsec_cxl_range_2_base_low

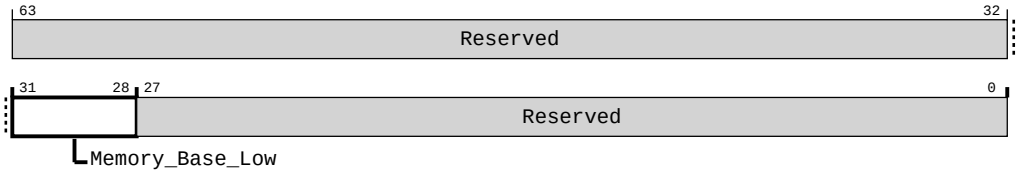


Table 4-210: por_cxlapb_dvsec_cxl_range_2_base_low attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	Memory_Base_Low	Corresponds to bits 31:28 of CXL Range 2 base in the host address space. Locked by CONFIG_LOCK	RWL	4'h0
[27:0]	Reserved	Reserved	RO	-

4.3.6.20 por_cxlapb_dvsec_cxl_control

Contains DVSEC_CXL_Control. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h10C`

Type

RWL

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-195: por_cxlapb_dvsec_cxl_control

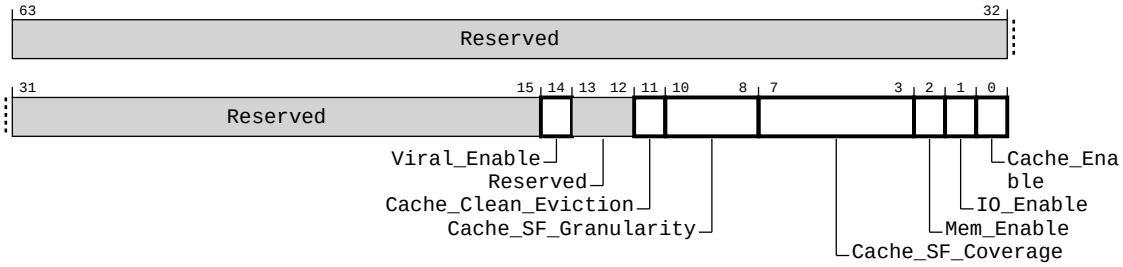


Table 4-211: por_cxlapb_dvsec_cxl_control attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14]	Viral_Enable	When set enables Viral handling in the CXL device. Locked by CONFIG_LOCK. If 0 the CXL device may ignore the viral that it receives	RWL	1'h0
[13:12]	Reserved	Reserved	RO	-
[11]	Cache_Clean_Eviction	Performance hint to the device. Locked by CONFIG_LOCK. **0** Indicates clean evictions from device caches are needed for best performance **1** Indicates clean evictions from device caches are NOT needed for best performance	RWL	1'h0
[10:8]	Cache_SF_Granularity	Performance hint to the device. Locked by CONFIG_LOCK. **000** Indicates 64B granular tracking on the Host **001** Indicates 128B granular tracking on the Host **010** Indicates 256B granular tracking on the Host **011** Indicates 512B granular tracking on the Host **100** Indicates 1KB granular tracking on the Host **101** Indicates 2KB granular tracking on the Host **110** Indicates 4KB granular tracking on the Host **111** Reserved	RWL	3'h0
[7:3]	Cache_SF_Coverage	Performance hint to the device. Locked by CONFIG_LOCK. 0x00: Indicates no Snoop Filter coverage on the Host For all other values of N: Indicates Snoop Filter coverage on the Host of 2^(N+15d) Bytes.	RWL	5'h0
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	1'h0
[1]	IO_Enable	When set enables CXL.io protocol operation when in Flex Bus.CXL mode.	RWL	1'h1
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode. Locked by CONFIG_LOCK.	RWL	1'h0

4.3.6.21 por_cxlapb_dvsec_cxl_control2

Contains DVSEC_CXL_Control2. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h110`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-196: por_cxlapb_dvsec_cxl_control2

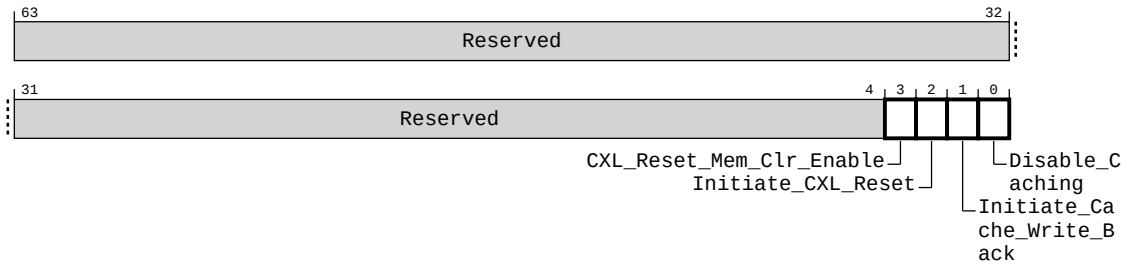


Table 4-212: por_cxlapb_dvsec_cxl_control2 attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	CXL_Reset_Mem_Clr_Enable	When set and CXL Reset Mem Clr Capable returns 1 Device shall clear or randomize volatile HDM ranges as part of the CXL Reset operation. When CXL Reset Mem Clr Capable is clear this bit is ignored and volatile HDM ranges may or may not be cleared or randomized during CXL Reset.	RW	1'h0
[2]	Initiate_CXL_Reset	When set to 1 device shall initiate CXL Reset as defined in Section 9.7. This bit always returns the value of 0 when read by the software. A write of 0 is ignored.	RW	1'h0
[1]	Initiate_Cache_Write_Back	When set to 1 device shall write back all modified lines in the local cache and invalidate all lines. The device shall send CacheFlushed message to host as required by CXL.Cache protocol to indicate it does not hold any modified lines.	RW	1'h0
[0]	Disable_Caching	When set to 1 device shall no longer cache new modified lines in its local cache. Device shall continue to correctly respond to CXL.cache transactions.	RW	1'h0

4.3.6.22 por_cxlapb_dvsec_cxl_lock

Contains DVSEC_CXL_Lock. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h114`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-197: por_cxlapb_dvsec_cxl_lock

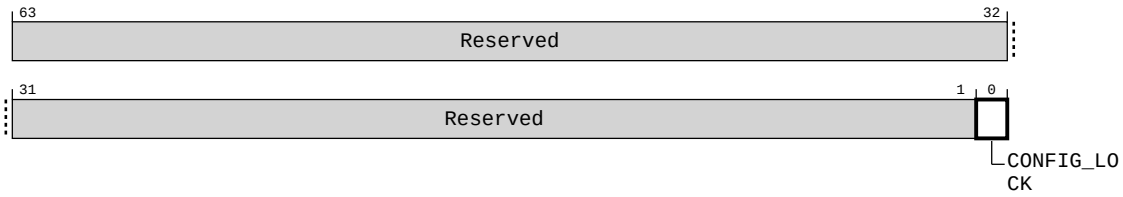


Table 4-213: por_cxlapb_dvsec_cxl_lock attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	CONFIG_LOCK	When set all register fields in the PCIe DVSEC for CXL Devices Capability with RWL attribute become read only. Consult individual register fields for details. This bit is cleared upon device Conventional Reset. This bit and all the fields that are locked by this bit are not affected by CXL Reset.	RW	1'h0

4.3.6.23 por_cxlapb_dvsec_flex_bus_port_control

Contains DVSEC_Flex_Bus_Port_Control

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h20C`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-198: por_cxlapb_dvsec_flex_bus_port_control

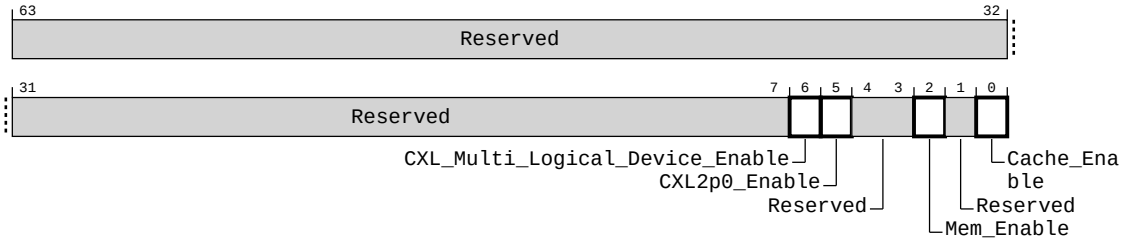


Table 4-214: por_cxlapb_dvsec_flex_bus_port_control attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	CXL_Multi_Logical_Device_Enable	When set enable Multi-Logical Device operation when in Flex Bus.CXL mode	RW	1'h0
[5]	CXL2p0_Enable	When set enable CXL2.0 protocol operation when in Flex Bus.CXL mode.	RW	1'h0
[4:3]	Reserved	Reserved	RO	-
[2]	Mem_Enable	When set enables CXL.mem protocol operation when in Flex Bus.CXL mode.	RW	1'h0
[1]	Reserved	Reserved	RO	-
[0]	Cache_Enable	When set enables CXL.cache protocol operation when in Flex Bus.CXL mode.	RW	1'h0

4.3.6.24 por_cxlapb_err_capabilities_control

Contains err_capabilities_control. Only applicable for Device. Host does not use this register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1014`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-199: por_cxlapb_err_capabilities_control

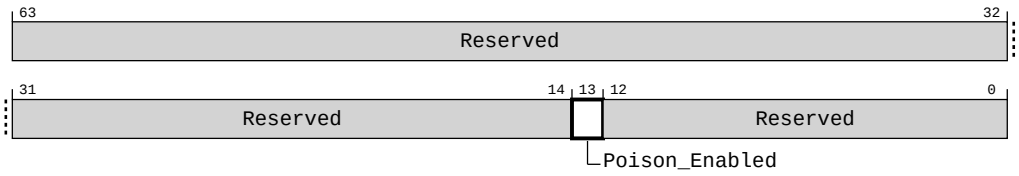


Table 4-215: por_cxlapb_err_capabilities_control attributes

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13]	Poison_Enabled	If this bit is 0 CXL 1.1 Upstream Ports CXL 1.1 Downstream Ports and CXL 2.0 Root Port shall treat poison received on CXL.cache or CXL.mem as uncorrectable error and log the error in Uncorrectable Error Status Register. If this bit is 1 these ports shall treat poison received on CXL.cache or CXL.mem as correctable error and log the error in Correctable Error Status Register. This bit defaults to 1. This bit is hardwired to 1 in CXL 2.0 Upstream Switch Port CXL 2.0 Downstream Switch Port and CXL 2.0 device.	RW	1'h0
[12:0]	Reserved	Reserved	RO	-

4.3.6.25 por_cxlapb_IDE_key_refresh_time_control

Contains IDE_key_refresh_time_control. Not applicable to CMN 700

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h18`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-200: por_cxlapb_IDE_key_refresh_time_control

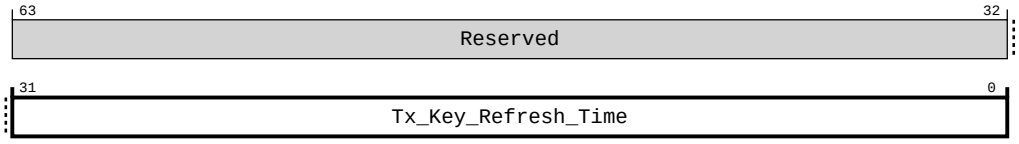


Table 4-216: por_cxlapb_IDE_key_refresh_time_control attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	Tx_Key_Refresh_Time	Minimum number of flits transmitter needs to block transmission of protocol flits after IDE.Start has sent. Used when switching keys.	RW	32'h0

4.3.6.26 por_cxlapb_IDE_truncation_transmit_delay_control

Contains IDE_truncation_transmit_delay_control. Not applicable to CMN 700

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h24

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-201: por_cxlapb_IDE_truncation_transmit_delay_control

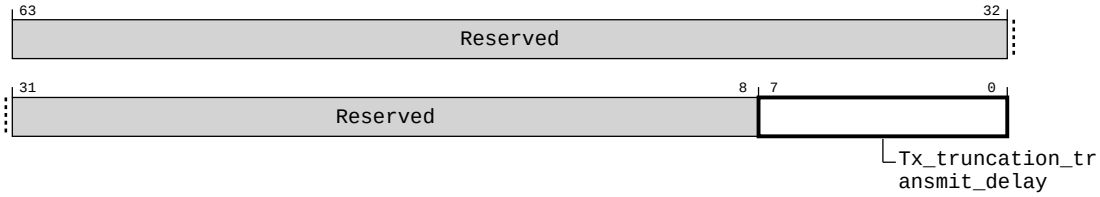


Table 4-217: por_cxlapb_IDE_truncation_transmit_delay_control attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:0]	Tx_truncation_transmit_delay	This parameter feeds into the computation of minimum number of IDE idle flits Transmitter needs send after sending a truncated MAC flit.	RW	8'h0

4.3.6.27 por_cxlapb_ll_to_ull_msg

Contains lll_to_ull_message

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h0

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-202: por_cxlapb_ll_to_ull_msg

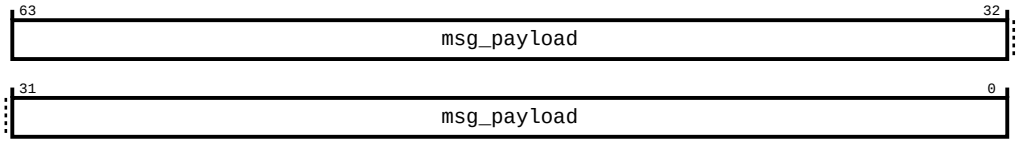


Table 4-218: por_cxlapb_ll_to_ull_msg attributes

Bits	Name	Description	Type	Reset
[63:0]	msg_payload	Contains 64 bits of message sent from ll to ull	RW	64'h0

4.3.6.28 por_cxlapb_cxl_timeout_isolation_control

Contains cxl_timeout_isolation_control. Not applicable to CMN 700

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h8`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-203: por_cxlapb_cxl_timeout_isolation_control

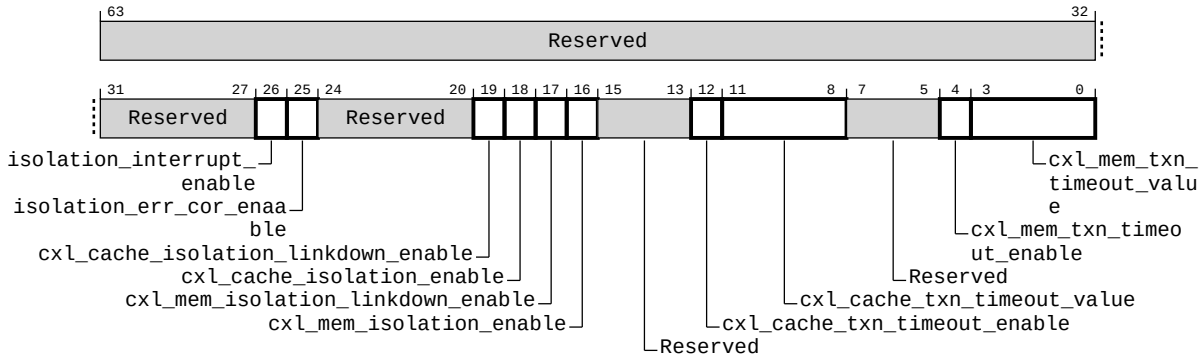


Table 4-219: por_cxlapb_cxl_timeout_isolation_control attributes

Bits	Name	Description	Type	Reset
[63:27]	Reserved	Reserved	RO	-
[26]	isolation_interrupt_enable	When Set this bit enables the generation of an interrupt to indicate that Isolation has been triggered.	RW	1'h0
[25]	isolation_err_cor_enaable	When Set this bit enables the sending of an ERR_COR Message to indicate Isolation has been triggered.	RW	1'h0
[24:20]	Reserved	Reserved	RO	-
[19]	cxl_cache_isolation_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.cache enters Isolation mode.	RW	1'h0
[18]	cxl_cache_isolation_enable	This field allows System Software to enable CXL.cache Isolation actions.	RW	1'h0
[17]	cxl_mem_isolation_linkdown_enable	This field allows System Software to trigger link down on the CXL Root Port if CXL.mem enters Isolation mode.	RW	1'h0
[16]	cxl_mem_isolation_enable	This field allows System Software to enable CXL.mem Isolation actions. If this field is set Isolation actions will be triggered if either a CXL.mem Transaction Timeout is detected or if the CXL link went down.	RW	1'h0
[15:13]	Reserved	Reserved	RO	-
[12]	cxl_cache_txn_timeout_enable	-	RW	1'h0
[11:8]	cxl_cache_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.cache.	RW	4'h0
[7:5]	Reserved	Reserved	RO	-
[4]	cxl_mem_txn_timeout_enable	When Set this bit enables CXL.mem Transaction Timeout mechanism.	RW	1'h0
[3:0]	cxl_mem_txn_timeout_value	In CXL Root Port Functions that support Transaction Timeout programmability this field allows system software to modify the Transaction Timeout Value for CXL.mem.	RW	4'h0

4.3.6.29 por_cxlapb_link_layer_defeature

CXL Link Layer Defeature Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1130`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-204: por_cxlapb_link_layer_defeature

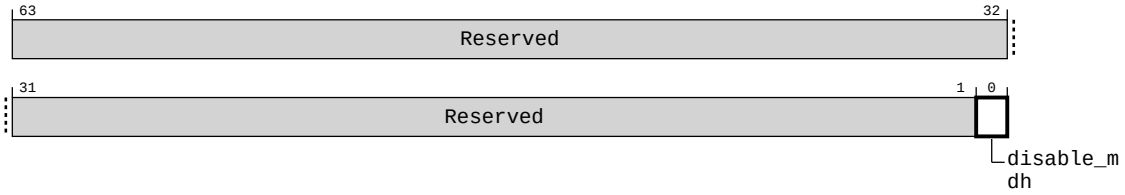


Table 4-220: por_cxlapb_link_layer_defeature attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	disable_mdh	Write 1 to disable MDH. Software needs to ensure it programs this value consistently on the UP & DP. After programming, a warm reset is required for the disable to take effect.	RW	1'b0

4.3.7 Debug and trace register descriptions

This section lists the debug and trace registers.

4.3.7.1 por_dt_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-205: por_dt_node_info

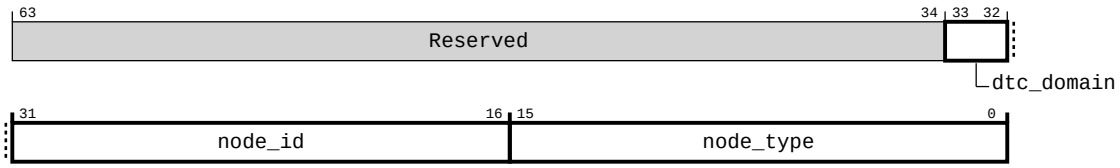


Table 4-221: por_dt_node_info attributes

Bits	Name	Description	Type	Reset
[63:34]	Reserved	Reserved	RO	-
[33:32]	dtc_domain	DTC domain number	RO	Configuration dependent
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h3

4.3.7.2 por_dt_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-206: por_dt_child_info

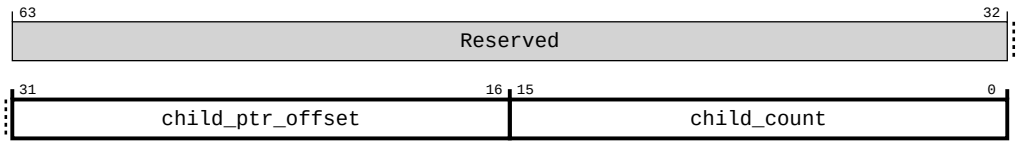


Table 4-222: por_dt_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0000
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.7.3 por_dt_secure_access

Functions as the Secure access control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-207: por_dt_secure_access

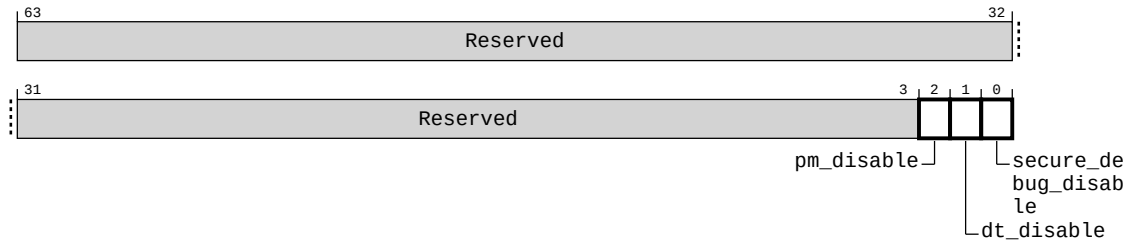


Table 4-223: por_dt_secure_access attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	pm_disable	PMU disable **1'b0** PMU function is not affected **1'b1** PMU function is disabled.	RW	1'b0
[1]	dt_disable	Debug disable **1'b0** DT function is not affected **1'b1** DT function is disabled.	RW	1'b0
[0]	secure_debug_disable	Secure debug disable **1'b0** Secure events are monitored by the PMU **1'b1** Secure events are only monitored by the PMU if SPNIDEN is set to 1	RW	1'b0

4.3.7.4 por_dt_dtc_ctl

Functions as the debug trace control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA00

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-208: por_dt_dtc_ctl

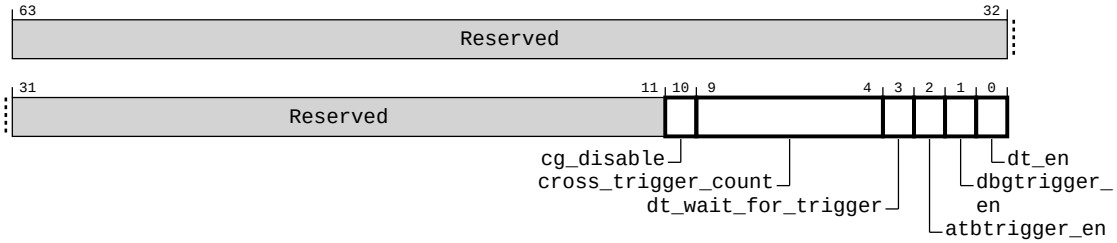


Table 4-224: por_dt_dtc_ctl attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	cg_disable	Disables DT architectural clock gates	RW	1'b0
[9:4]	cross_trigger_count	Number of cross triggers received before trace enable NOTE: Only applicable if dt_wait_for_trigger is set to 1.	RW	6'b0
[3]	dt_wait_for_trigger	Enables waiting for cross trigger before trace enable	RW	1'b0
[2]	atbtrigger_en	ATB trigger enable	RW	1'b0
[1]	dbgtrigger_en	DBGWATCHTRIG enable	RW	1'b0
[0]	dt_en	Enables debug, trace, and PMU features	RW	1'b0

4.3.7.5 por_dt_trigger_status

Provides the trigger status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA10`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-209: por_dt_trigger_status

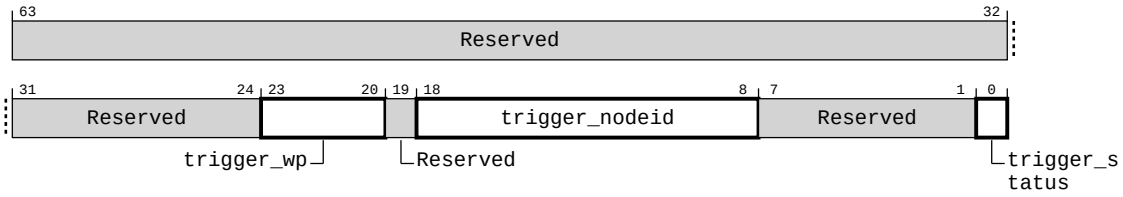


Table 4-225: por_dt_trigger_status attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	trigger_wp	DBGWATCHTRIGREQ assertion and/or ATB trigger are caused by watchpoint	RO	1'h0
[19]	Reserved	Reserved	RO	-
[18:8]	trigger_nodeid	DBGWATCHTRIGREQ assertion and/or ATB trigger are caused by node ID	RO	11'h0
[7:1]	Reserved	Reserved	RO	-
[0]	trigger_status	Indicates DBGWATCHTRIGREQ assertion and/or ATB trigger	RO	1'h0

4.3.7.6 por_dt_trigger_status_clr

Clears the trigger status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA20`

Type

WO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-210: por_dt_trigger_status_clr

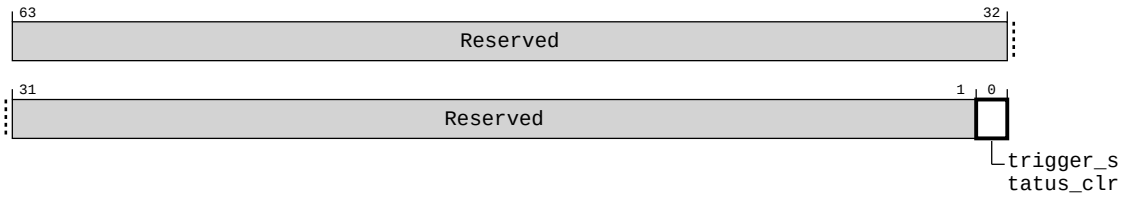


Table 4-226: por_dt_trigger_status_clr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	trigger_status_clr	Write a 1 to clear por_dt_trigger_status.trigger_status	WO	1'b0

4.3.7.7 por_dt_trace_control

Functions as the trace control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA30`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-211: por_dt_trace_control

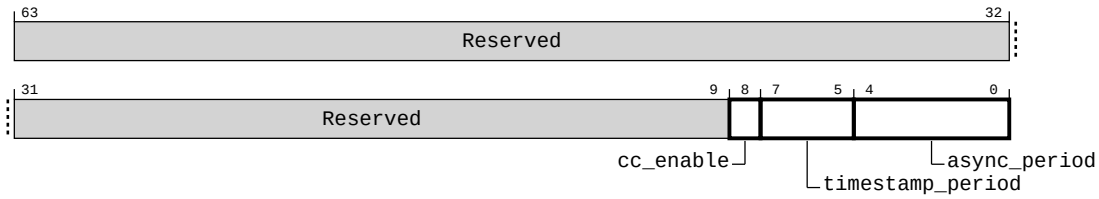


Table 4-227: por_dt_trace_control attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	cc_enable	Cycle count enable	RW	1'b0
[7:5]	timestamp_period	Time stamp packet insertion period 3'b000: Time stamp disabled 3'b011: Time stamp every 8K clock cycles 3'b100: Time stamp every 16K clock cycles 3'b101: Time stamp every 32K clock cycles 3'b110: Time stamp every 64K clock cycles	RW	3'b0
[4:0]	async_period	Alignment sync packet insertion period 5'h00: Alignment sync disabled 5'h08: Alignment sync inserted after 256B of trace 5'h09: Alignment sync inserted after 512B of trace 5'h14: Alignment sync inserted after 1048576B of trace NOTE: All other values are reserved.	RW	5'b0

4.3.7.8 por_dt_traceid

Contains the ATB ID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA48`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-212: por_dt_traceid

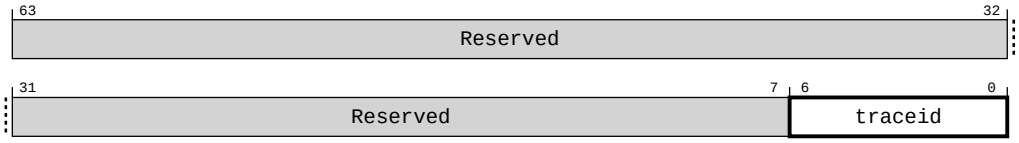


Table 4-228: por_dt_traceid attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6:0]	traceid	ATB ID	RW	7'h0

4.3.7.9 por_dt_pmevcntAB

Contains the PMU event counters A and B.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-213: por_dt_pmevcntAB

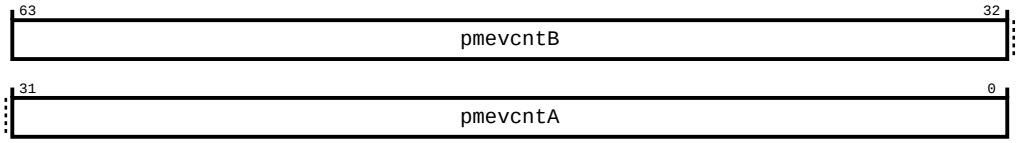


Table 4-229: por_dt_pmevcntAB attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntB	PMU counter B	RW	32'h0000
[31:0]	pmevcntA	PMU counter A	RW	32'h0000

4.3.7.10 por_dt_pmevcntCD

Contains the PMU event counters C and D.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2010`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-214: por_dt_pmevcntCD

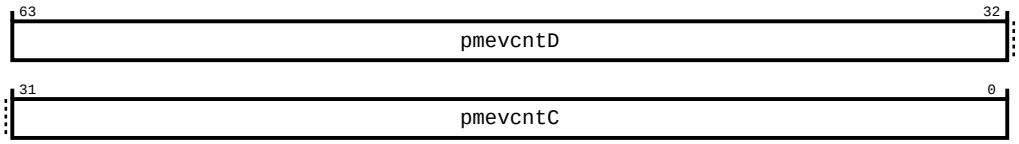


Table 4-230: por_dt_pmevcntCD attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntD	PMU counter D	RW	32'h0000
[31:0]	pmevcntC	PMU counter C	RW	32'h0000

4.3.7.11 por_dt_pmevcntEF

Contains the PMU event counters E and F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2020`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-215: por_dt_pmevcntEF

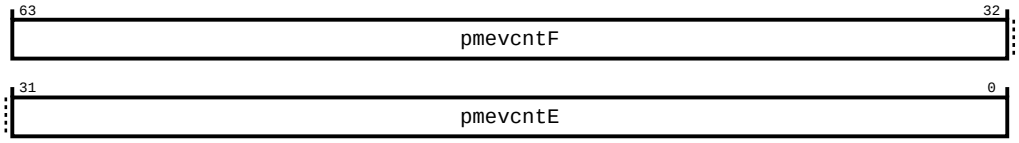


Table 4-231: por_dt_pmevcntEF attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntF	PMU counter F	RW	32'h0000
[31:0]	pmevcntE	PMU counter E	RW	32'h0000

4.3.7.12 por_dt_pmevcntGH

Contains the PMU event counters G and H.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2030`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-216: por_dt_pmevcntGH

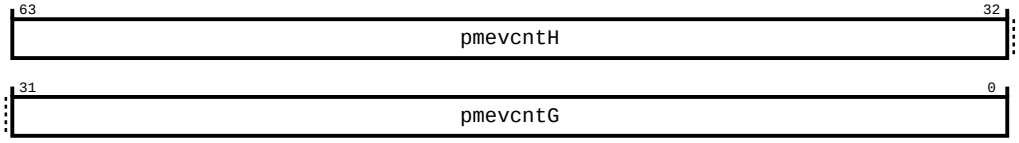


Table 4-232: por_dt_pmevcntGH attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntH	PMU counter H	RW	32'h0000
[31:0]	pmevcntG	PMU counter G	RW	32'h0000

4.3.7.13 por_dt_pmccntr

Contains the PMU cycle counter.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2040`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-217: por_dt_pmccntr

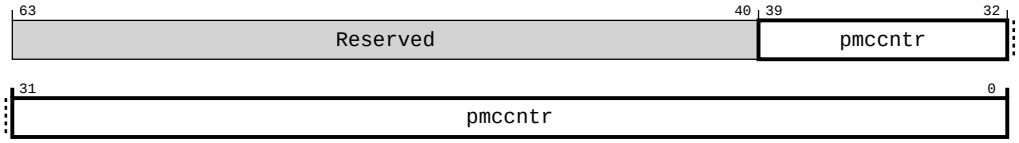


Table 4-233: por_dt_pmccntr attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:0]	pmccntr	PMU cycle counter	RW	40'h0

4.3.7.14 por_dt_pmevcntsrAB

Contains the PMU event counter shadow registers A and B.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2050`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-218: por_dt_pmevcntsrAB

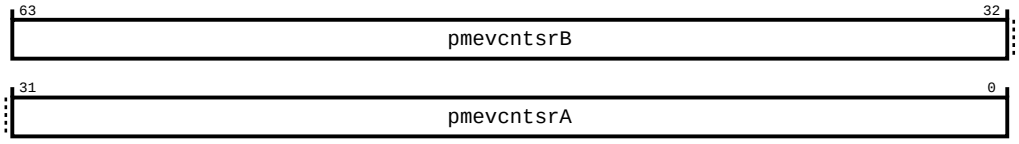


Table 4-234: por_dt_pmevcntsrAB attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrB	PMU counter B shadow register	RW	32'h0000
[31:0]	pmevcntsrA	PMU counter A shadow register	RW	32'h0000

4.3.7.15 por_dt_pmevcntsrCD

Contains the PMU event counter shadow registers C and D.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2060`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-219: por_dt_pmevcntsrCD

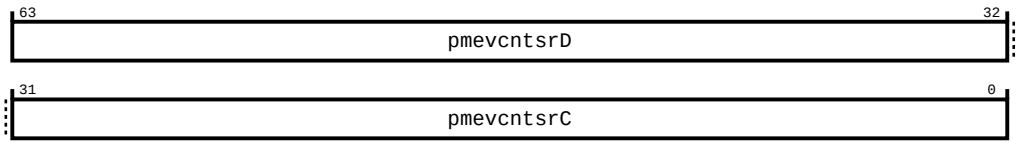


Table 4-235: por_dt_pmevcntsrCD attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrD	PMU counter D shadow register	RW	32'h0000
[31:0]	pmevcntsrC	PMU counter C shadow register	RW	32'h0000

4.3.7.16 por_dt_pmevcntsrEF

Contains the PMU event counter shadow registers E and F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2070`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-220: por_dt_pmevcntsrEF

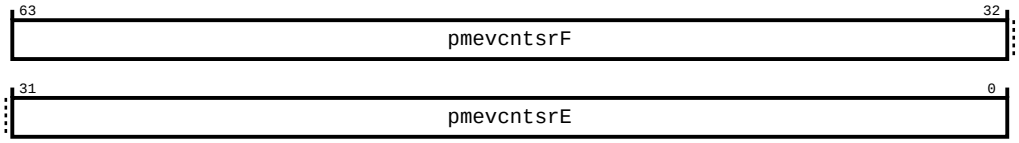


Table 4-236: por_dt_pmevcntsrEF attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrF	PMU counter F shadow register	RW	32'h0000
[31:0]	pmevcntsrE	PMU counter E shadow register	RW	32'h0000

4.3.7.17 por_dt_pmevcntsrGH

Contains the PMU event counter shadow registers G and H.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2080`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-221: por_dt_pmevcntsrGH

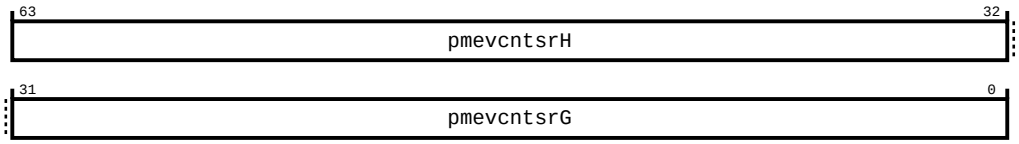


Table 4-237: por_dt_pmevcntsrGH attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntsrH	PMU counter H shadow register	RW	32'h0000
[31:0]	pmevcntsrG	PMU counter G shadow register	RW	32'h0000

4.3.7.18 por_dt_pmccntrsr

Contains the PMU cycle counter shadow register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2090`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-222: por_dt_pmccntrsr

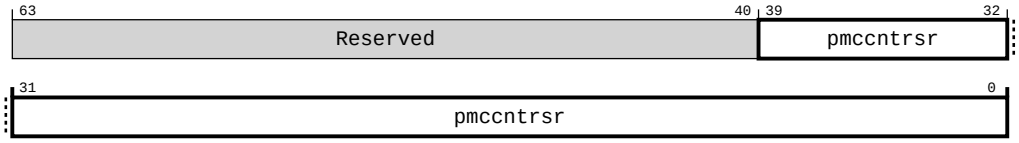


Table 4-238: por_dt_pmccntrsr attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:0]	pmccntrsr	PMU cycle counter shadow register	RW	40'h0

4.3.7.19 por_dt_pmcr

Functions as the PMU control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2100`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-223: por_dt_pmcr

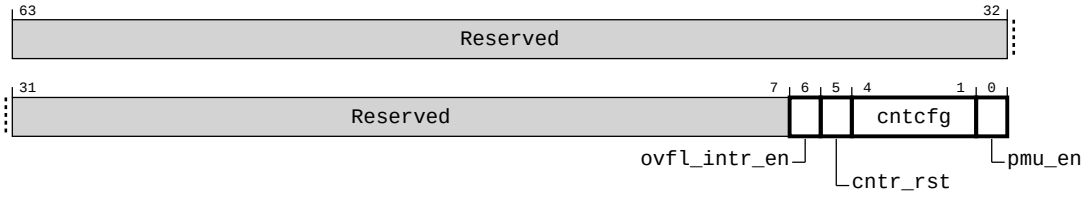


Table 4-239: por_dt_pmcr attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	ovfl_intr_en	Enables INTREQPMU assertion on PMU counter overflow	RW	1'h0
[5]	cntr_rst	Enables clearing of live counters upon assertion of por_dt_pmsrr.ss_req or PMUSNAPSHOTREQ	RW	1'h0
[4:1]	cntcfg	Groups adjacent 32-bit registers into a 64-bit register	RW	4'h0
[0]	pmu_en	Enables PMU features	RW	1'b0

4.3.7.20 por_dt_pmovsr

Provides the PMU overflow status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2118`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-224: por_dt_pmovsr

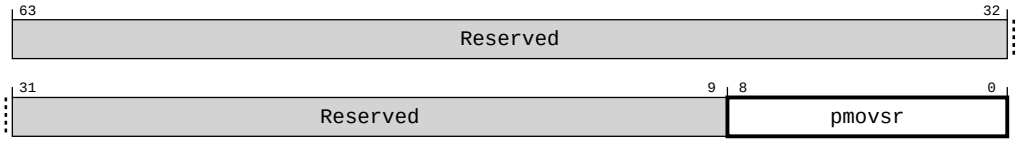


Table 4-240: por_dt_pmovsr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8:0]	pmovsr	PMU overflow status Bit 8: Indicates overflow from cycle counter Bits [7:0]: Indicates overflow from counters 7 to 0	RO	9'h0

4.3.7.21 por_dt_pmovsr_clr

Clears the PMU overflow status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2120`

Type

WO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-225: por_dt_pmovsr_clr

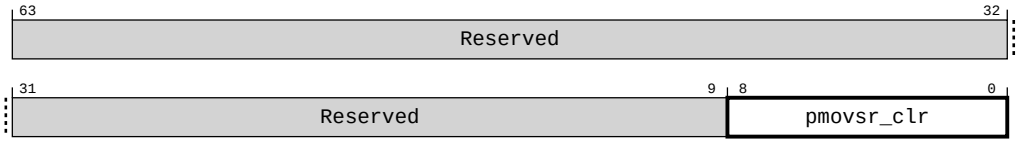


Table 4-241: por_dt_pmovsr_clr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8:0]	pmovsr_clr	Write a 1 to clear the corresponding bit in por_dt_pmovsr.pmovsr	WO	9'b0

4.3.7.22 por_dt_pmssr

Provides the PMU snapshot status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2128`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-226: por_dt_pmssr

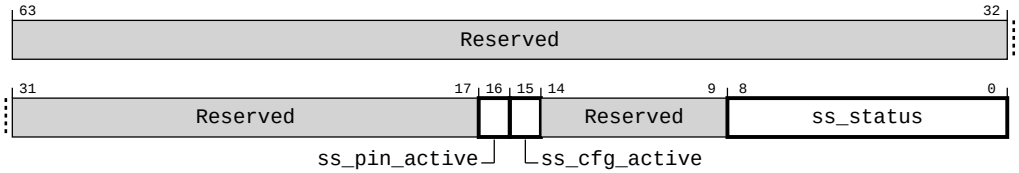


Table 4-242: por_dt_pmssr attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	ss_pin_active	Activates PMU snapshot from PMUSNAPSHOTREQ	RO	1'b0
[15]	ss_cfg_active	PMU snapshot activated from configuration write	RO	1'b0
[14:9]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[8:0]	ss_status	PMU snapshot status Bit 8: Indicates snapshot status for cycle counter Bits [7:0]: Indicates snapshot status for counters 7 to 0	RO	9'b0

4.3.7.23 por_dt_pmsrr

Sends PMU snapshot requests.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2130`

Type

WO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-227: por_dt_pmsrr

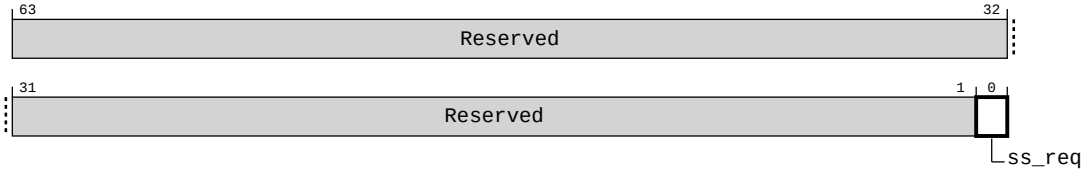


Table 4-243: por_dt_pmsrr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	ss_req	Write a 1 to request PMU snapshot	WO	1'b0

4.3.7.24 por_dt_claim

Functions as the claim tag set register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFA0`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-228: por_dt_claim

Table 4-244: por_dt_claim attributes

Bits	Name	Description	Type	Reset
[63:32]	clr	Upper half of the claim tag value; enables individual bits to be cleared (write) and returns the current claim tag value (read)	RW	32'b0
[31:0]	set	Lower half of the claim tag value; allows individual bits to be set (write) and returns the number of bits that can be set (read)	RW	32'hffffffff

4.3.7.25 por_dt_devaff

Functions as the device affinity register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFA8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-229: por_dt_devaff

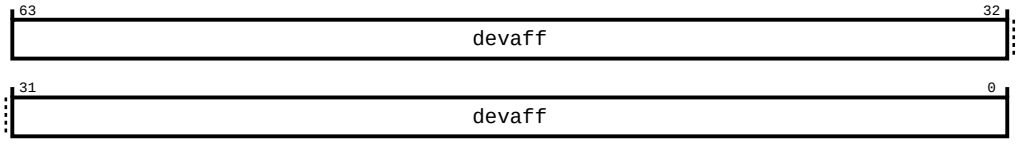


Table 4-245: por_dt_devaff attributes

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	64'b0

4.3.7.26 por_dt_lsr

Functions as the lock status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFB0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-230: por_dt_lsr

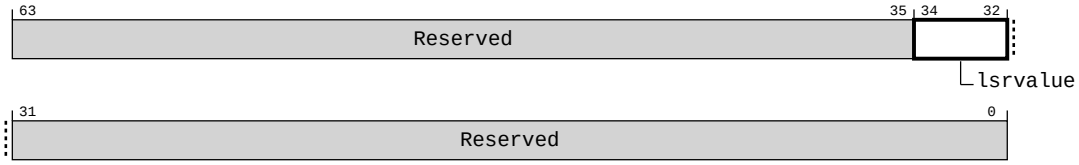


Table 4-246: por_dt_lsr attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34:32]	lsrvalue	Lock status value	RO	3'b0
[31:0]	Reserved	Reserved	RO	-

4.3.7.27 por_dt_authstatus_devarch

Functions as the authentication status register and the device architecture register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFB8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-231: por_dt_authstatus_devarch

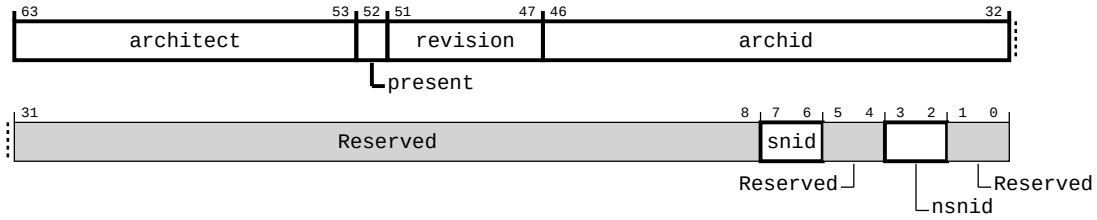


Table 4-247: por_dt_authstatus_devarch attributes

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	11'b0
[52]	present	Present	RO	1'b1
[51:47]	revision	Architecture revision	RO	6'b0
[46:32]	archid	Architecture ID	RO	16'b0
[31:8]	Reserved	Reserved	RO	-
[7:6]	snid	Secure non-invasive debug	RO	2'b10
[5:4]	Reserved	Reserved	RO	-
[3:2]	nsnid	Non-secure non-invasive debug	RO	2'b10
[1:0]	Reserved	Reserved	RO	-

4.3.7.28 por_dt_devid

Functions as the device configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFC0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-232: por_dt_devid

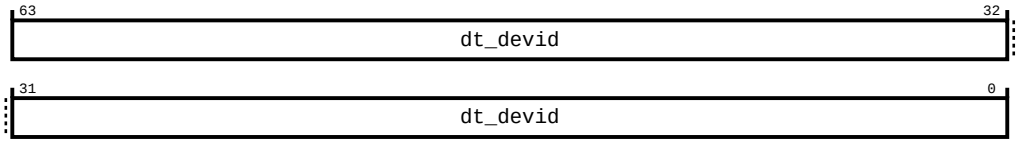


Table 4-248: por_dt_devid attributes

Bits	Name	Description	Type	Reset
[63:0]	dt_devid	Device ID	RO	64'b0

4.3.7.29 por_dt_devtype

Functions as the device type identifier register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFC8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-233: por_dt_devtype

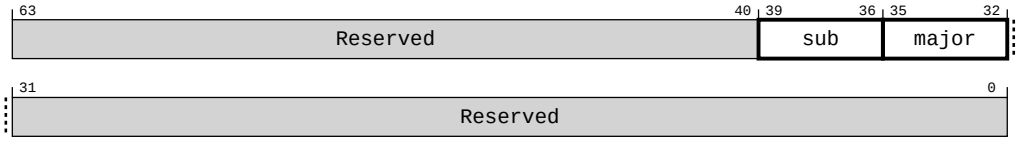


Table 4-249: por_dt_devtype attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:36]	sub	Sub type	RO	4'h4
[35:32]	major	Major type	RO	4'h3
[31:0]	Reserved	Reserved	RO	-

4.3.7.30 por_dt_pidr45

Functions as the identification register for peripheral ID 4 and peripheral ID 5.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFD0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-234: por_dt_pidr45

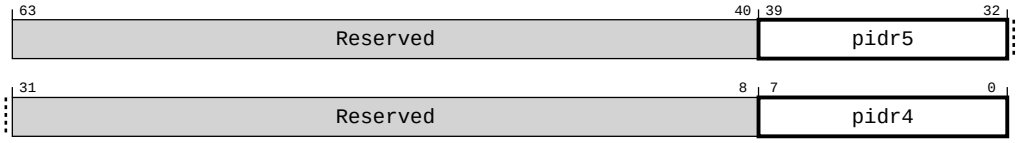


Table 4-250: por_dt_pidr45 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr5	Peripheral ID 5	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr4	Peripheral ID 4	RO	8'h4

4.3.7.31 por_dt_pidr67

Functions as the identification register for peripheral ID 6 and peripheral ID 7.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFD8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-235: por_dt_pidr67

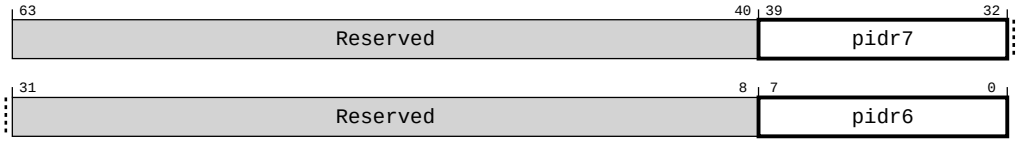


Table 4-251: por_dt_pidr67 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr7	Peripheral ID 7	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr6	Peripheral ID 6	RO	8'b0

4.3.7.32 por_dt_pidr01

Functions as the identification register for peripheral ID 0 and peripheral ID 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFE0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-236: por_dt_pidr01

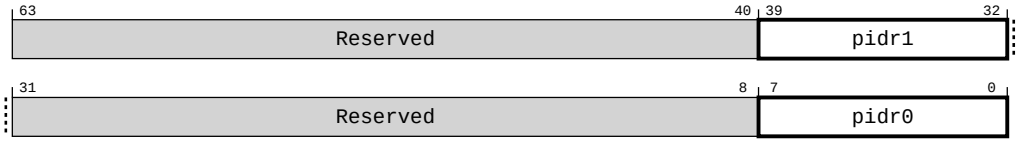


Table 4-252: por_dt_pidr01 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr1	Peripheral ID 1	RO	8'hb4
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr0	Peripheral ID 0	RO	8'h34

4.3.7.33 por_dt_pidr23

Functions as the identification register for peripheral ID 2 and peripheral ID 3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFE8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-237: por_dt_pidr23

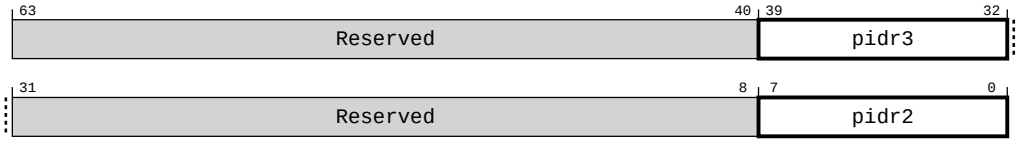


Table 4-253: por_dt_pidr23 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pidr3	Peripheral ID 3	RO	8'b0
[31:8]	Reserved	Reserved	RO	-
[7:0]	pidr2	Peripheral ID 2	RO	8'h7

4.3.7.34 por_dt_cidr01

Functions as the identification register for component ID 0 and component ID 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFF0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-238: por_dt_cidr01

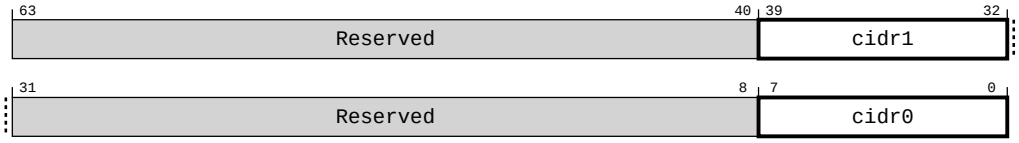


Table 4-254: por_dt_cidr01 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	cidr1	Component ID 1	RO	8'h9f
[31:8]	Reserved	Reserved	RO	-
[7:0]	cidr0	Component ID 0	RO	8'hd

4.3.7.35 por_dt_cidr23

Functions as the identification register for component ID 2 and component ID 3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFF8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-239: por_dt_cidr23

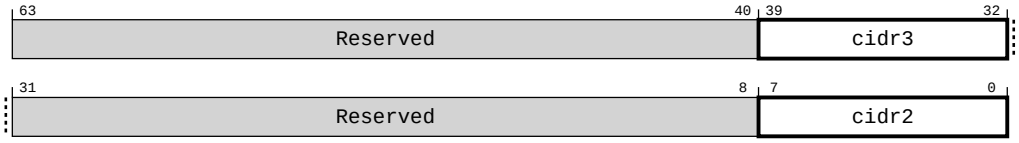


Table 4-255: por_dt_cidr23 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	cidr3	Component ID 3	RO	8'hb1
[31:8]	Reserved	Reserved	RO	-
[7:0]	cidr2	Component ID 2	RO	8'h5

4.3.8 DN register descriptions

This section lists the DN registers.

4.3.8.1 por_dn_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-240: por_dn_node_info

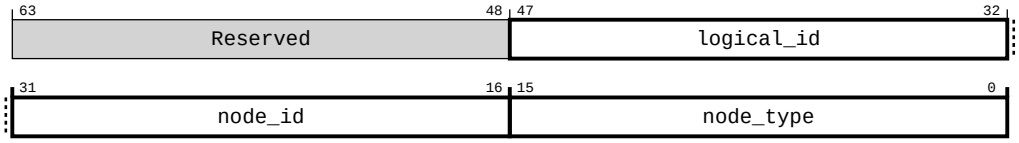


Table 4-256: por_dn_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component CHI node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0001

4.3.8.2 por_dn_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-241: por_dn_child_info

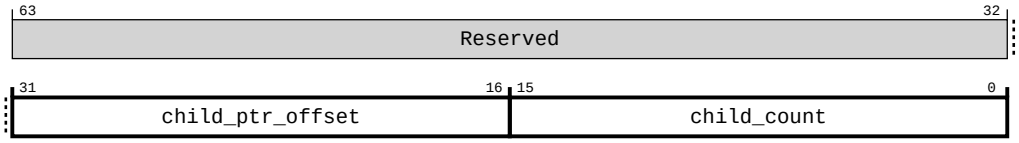


Table 4-257: por_dn_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'h0

4.3.8.3 por_dn_build_info

Contains the configuration parameter values. Indicates the specific DN configuration.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-242: por_dn_build_info

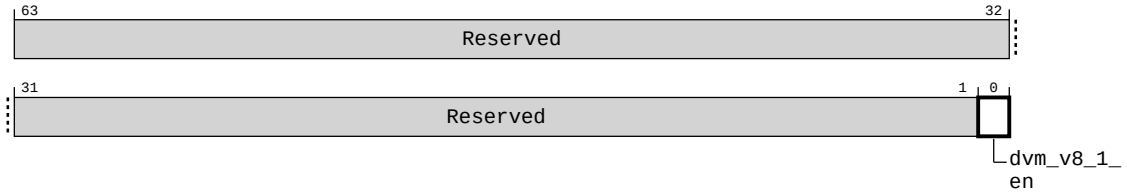


Table 4-258: por_dn_build_info attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	dvm_v8_1_en	Indicates that all nodes receiving DVM snoops support DVM v8/v8.1 operations.	RO	1'b1

4.3.8.4 por_dn_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-243: por_dn_secure_register_groups_override

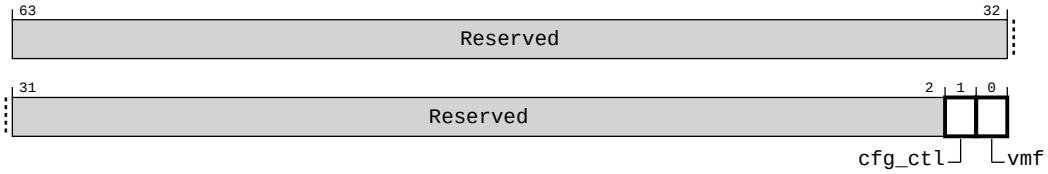


Table 4-259: por_dn_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	cfg_ctl	Allows Non-secure access to Secure configuration control register (por_dn_cfg_ctl)	RW	1'b0
[0]	vmf	Allows Non-secure access to Secure VMF registers	RW	1'b0

4.3.8.5 por_dn_cfg_ctl

Functions as the configuration control register for DVM Node.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-244: por_dn_cfg_ctl

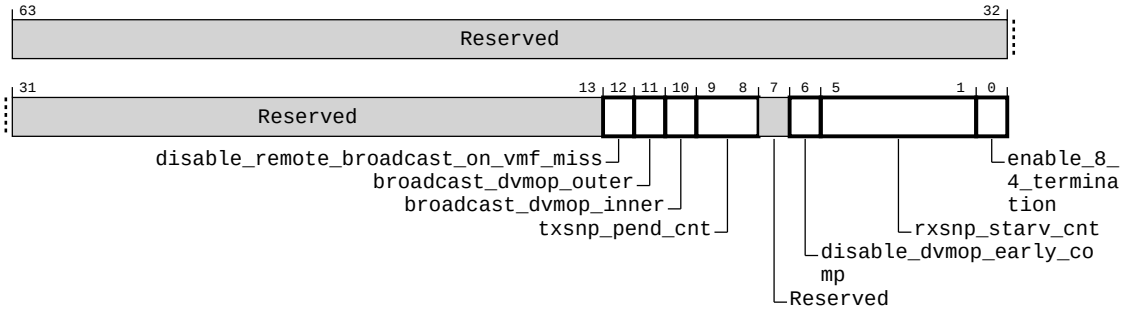


Table 4-260: por_dn_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:13]	Reserved	Reserved	RO	-
[12]	disable_remote_broadcast_on_vmf_miss	Disables broadcast of VMID Filterable DVMOps to remote on VMF miss.	RW	1'h0
[11]	broadcast_dvmop_outer	Used to filter DVMOps marked as outersharable (OS) from being sent off-chip.	RW	1'h1
[10]	broadcast_dvmop_inner	Used to filter DVMOps marked as innersharable (IS) from being sent off-chip.	RW	1'h1
[9:8]	txsnp_pend_cnt	Maximum number of (Non-Sync + Sync) SnpDVMOps issued on TXSNP. 2'b00: Max of 4 SnpDVMOps in progress (default) 2'b01: Max of 8 SnpDVMOps in progress 2'b10: Max of 16 SnpDVMOps in progress 2'b11: Reserved	RW	2'h0
[7]	Reserved	Reserved	RO	-
[6]	disable_dvmop_early_comp	Disables Early Comp (CompDBID) for DVMOps	RW	1'b0
[5:1]	rxsnp_starv_cnt	Number of cycles RXSNP lost to RXREQ for RCB alloc.	RW	5'h8
[0]	enable_8_4_termination	Enables termination of 8.4 DVMOps in DN.	RW	1'b0

4.3.8.6 por_dn_aux_ctl

Functions as the auxiliary control register for DN.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-245: por_dn_aux_ctl

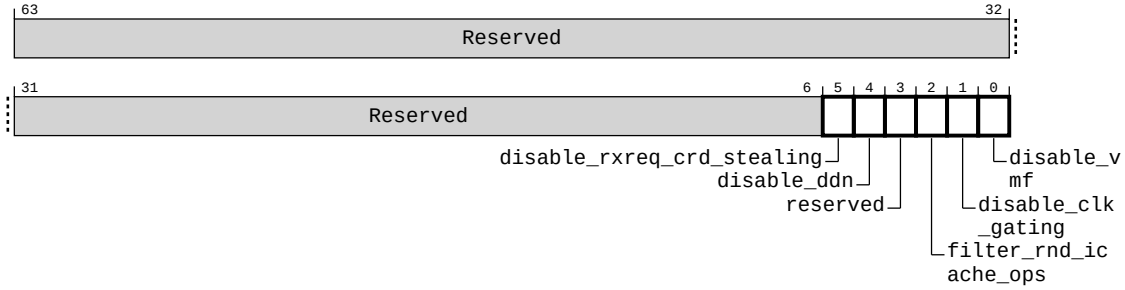


Table 4-261: por_dn_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5]	disable_rxreq_crd_stealing	Disables credit stealing from RXREQ LinkLayer when RXSNP is starved for RCB alloc.	RW	1'b0
[4]	disable_ddn	Disables Distributed DN functionality- Snoops all RNs and CML nodes in the mesh and disables snooping other DNs. Must program all RNSAMs to target HND for DVMs and then set this to 1 in HND.	RW	1'b0
[3]	reserved	Reserved field	RW	1'b0
[2]	filter_rnd_icache_ops	Filters out BPI and VICI/PICI Snps to RNDs when set	RW	Configuration dependent
[1]	disable_clk_gating	Disables autonomous clock gating when set	RW	1'b0
[0]	disable_vmf	Disables VMID-based DVM snoop filtering when set	RW	Configuration dependent

4.3.8.7 por_dn_vmf0-15_ctrl

There are 16 iterations of this register. The index ranges from 0 to 15. Functions as the control register for VMID-based DVM snoop filtering. NOTE: This register has no effect when por_dn_aux_ctl.disable_vmf is set to 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-246: por_dn_vmf0-15_ctrl

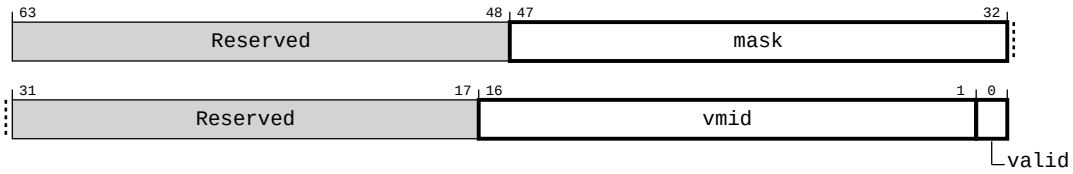


Table 4-262: por_dn_vmf0-15_ctrl attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	mask	VMID mask; enables mapping of multiple VMID values to a single register NOTE: Logically, the AND operator is performed on the mask and por_dn_vmf#{index}_ctrl.vmid. Then, the AND operator is performed on the mask and the incoming request's VMID. The two results are then compared, and filtering is applied to the incoming request if the masked VMIDs match.	RW	16'hffff
[31:17]	Reserved	Reserved	RO	-
[16:1]	vmid	VMID value NOTE: The incoming request's VMID is only compared with this VMID value if the request's VMID valid bit is set. If the request's VMID is valid and the two VMIDs match, filtering is applied to the incoming request.	RW	16'h0000
[0]	valid	Register valid 1'b1: Register is enabled 1'b0: Register is not enabled	RW	1'b0

4.3.8.8 por_dn_vmf0-15_rnf0

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index+8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-247: por_dn_vmf0-15_rnf0

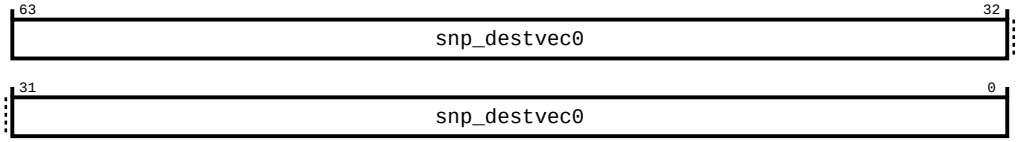


Table 4-263: por_dn_vmf0-15_rnf0 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0	RN-F bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.9 por_dn_vmf0-15_rnf1

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index+16}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-248: por_dn_vmf0-15_rnf1

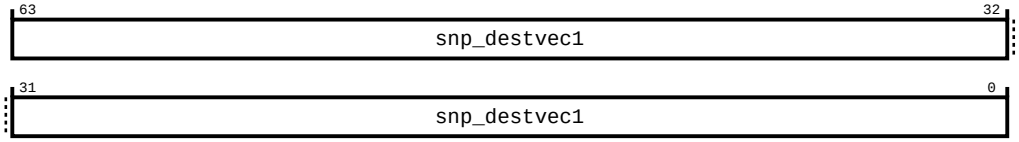


Table 4-264: por_dn_vmf0-15_rnf1 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1	RN-F bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.10 por_dn_vmf0-15_rnf2

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 191:128 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index+24}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-249: por_dn_vmf0-15_rnf2

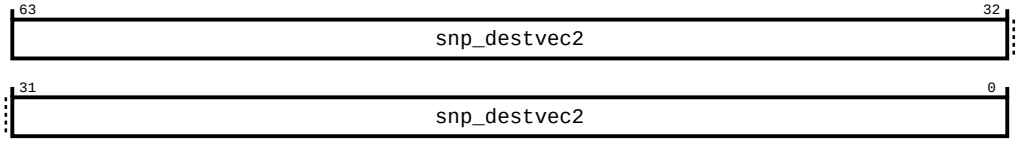


Table 4-265: por_dn_vmf0-15_rnf2 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec2	RN-F bit vector 191:128 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.11 por_dn_vmf0-15_rnf3

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-F bit vector 255:192 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index+32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-250: por_dn_vmf0-15_rnf3

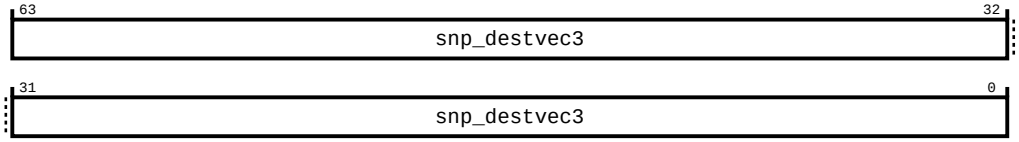


Table 4-266: por_dn_vmf0-15_rnf3 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec3	RN-F bit vector 255:192 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.12 por_dn_vmf0-15_rnd0

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-D bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index+40}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-251: por_dn_vmf0-15_rnd0

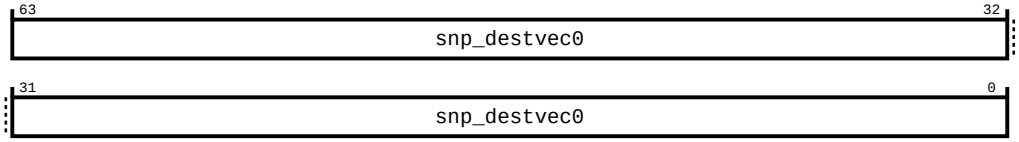


Table 4-267: por_dn_vmf0-15_rnd0 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0	RN-D bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.13 por_dn_vmf0-15_cxra

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical CXRA bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering. NOTE: Not applicable in a single-chip CMN-700 system. Does not have any effect.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{56*index+48}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-252: por_dn_vmf0-15_cxra

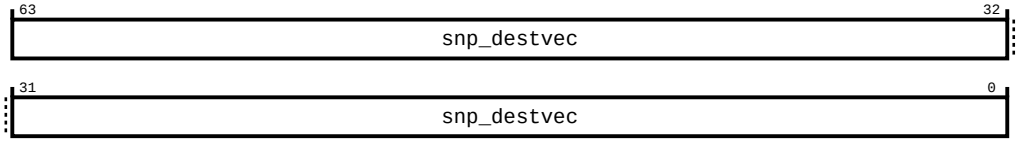


Table 4-268: por_dn_vmf0-15_cxra attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec	CXRA bit vector 63:0 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.14 por_dn_domain_rnf0-3

There are 4 iterations of this register. The index ranges from 0 to 3. RNF logical list for DDN

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF80 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-253: por_dn_domain_rnf0-3

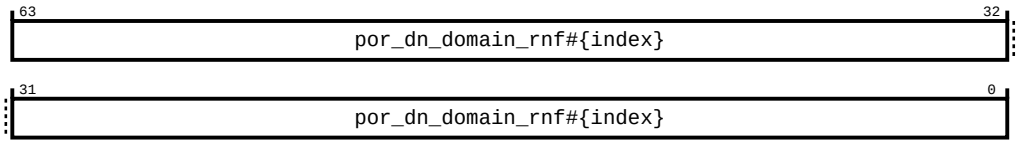


Table 4-269: por_dn_domain_rnf0-3 attributes

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_rnf#{index}	RNF logical list corresponding to RNF #{((index +1)*64)-1}:#{index*64}	RW	Configuration dependent

4.3.8.15 por_dn_domain_rnd0

RND logical list for DDN

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFA0`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-254: por_dn_domain_rnd0

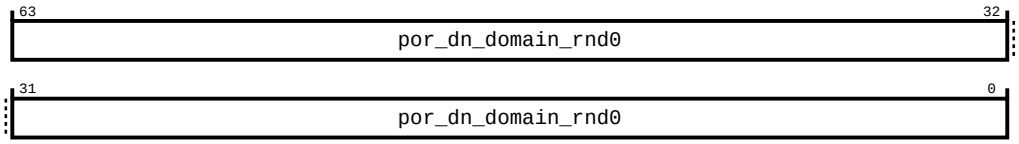


Table 4-270: por_dn_domain_rnd0 attributes

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_rnd0	RND logical list for DDN corresponding to RND 63:0	RW	Configuration dependent

4.3.8.16 por_dn_domain_cxra

CXRA logical list for DDN

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFA8`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-255: por_dn_domain_cxra

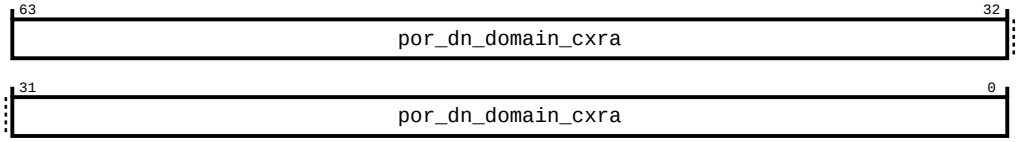


Table 4-271: por_dn_domain_cxra attributes

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_cxra	CXRA logical list for DDN	RW	Configuration dependent

4.3.8.17 por_dn_vmf0-15_rnd1

There are 16 iterations of this register. The index ranges from 0 to 15. Contains the logical RN-D bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid. Used for VMID-based DVM snoop filtering.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFB0 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_dn_secure_register_groups_override.vmf

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-256: por_dn_vmf0-15_rnd1

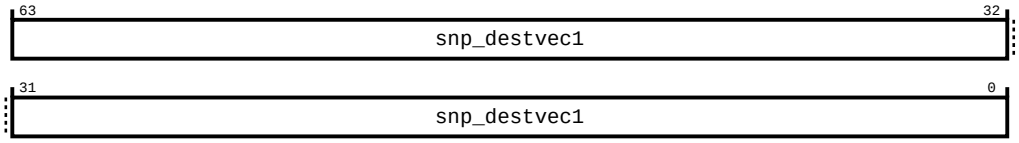


Table 4-272: por_dn_vmf0-15_rnd1 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1	RN-D bit vector 127:64 corresponding to por_dn_vmf#{index}_ctrl.vmid	RW	64'b0

4.3.8.18 por_dn_domain_rnd1

RND logical list for DDN

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1030`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-257: por_dn_domain_rnd1

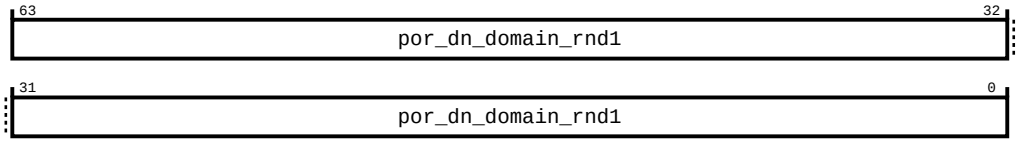


Table 4-273: por_dn_domain_rnd1 attributes

Bits	Name	Description	Type	Reset
[63:0]	por_dn_domain_rnd1	RND logical list for DDN corresponding to RND 127:64	RW	Configuration dependent

4.3.8.19 por_dn_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-258: por_dn_pmu_event_sel

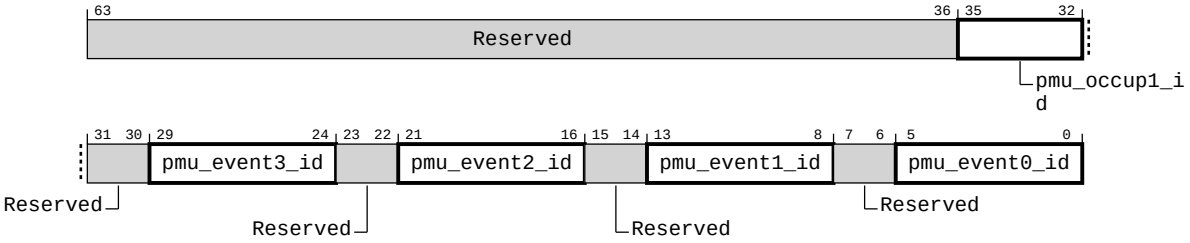


Table 4-274: por_dn_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	-
[35:32]	pmu_occup1_id	PMU occupancy event selector ID **4'b0000** All **4'b0001** DVM ops **4'b0010** DVM syncs	RW	4'b0
[31:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	PMU Event 3 ID; see pmu_event0_id for encodings	RW	5'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	PMU Event 2 ID; see pmu_event0_id for encodings	RW	5'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	PMU Event 1 ID; see pmu_event0_id for encodings	RW	5'b0
[7:6]	Reserved	Reserved	RO	-
[5:0]	pmu_event0_id	PMU Event 0 ID **6'h00** No event **6'h01** Number of TLBI DVM op requests **6'h02** Number of BPI DVM op requests **6'h03** Number of PICI DVM op requests **6'h04** Number of VICI DVM op requests **6'h05** Number of DVM sync requests **6'h06** Number of DVM op requests that were filtered using VMID filtering **6'h07** Number of DVM op requests to RNDs, BPI or PICI/VICI, that were filtered **6'h08** Number of retried REQ **6'h09** Number of SNPs sent to RNs **6'h0a** Number of SNPs stalled to RNs due to lack of Crds **6'h0b** DVM tracker full counter **6'h0c** DVM RNF tracker occupancy counter **6'h0d** DVM CXHA tracker occupancy counter **6'h0e** DVM Peer DN tracker occupancy counter **6'h0f** DVM RNF tracker Alloc **6'h10** DVM CXHA tracker Alloc **6'h11** DVM Peer DN tracker Alloc **6'h12** TXSNP stall due to number outstanding limit **6'h13** RXSNP stall starvation threshold hit **6'h14** TXSNP SYNC stall due to outstanding early completed Op	RW	5'b0

4.3.9 HN-I register descriptions

This section lists the HN-I registers.

4.3.9.1 por_hni_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-259: por_hni_node_info

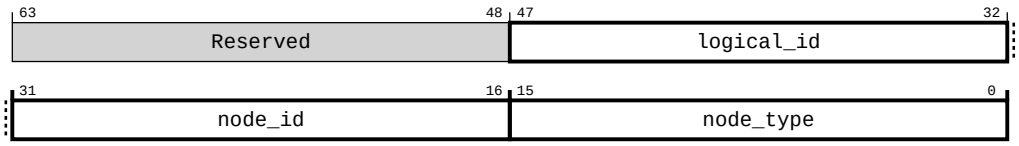


Table 4-275: por_hni_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	Configuration dependent

4.3.9.2 por_hni_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-260: por_hni_child_info

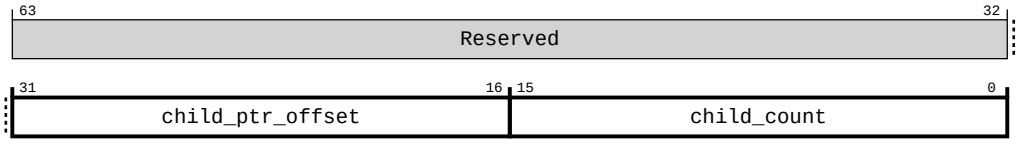


Table 4-276: por_hni_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.9.3 por_hni_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-261: por_hni_secure_register_groups_override

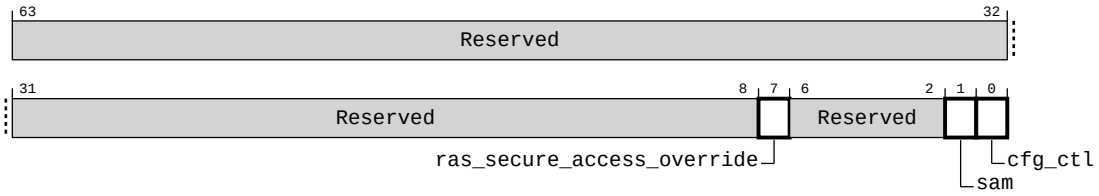


Table 4-277: por_hni_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7]	ras_secure_access_override	Allow Non-secure access to Secure RAS registers	RW	1'b0
[6:2]	Reserved	Reserved	RO	-
[1]	sam	Allows Non-secure access to Secure SAM registers	RW	1'b0
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register	RW	1'b0

4.3.9.4 por_hni_unit_info

Provides component identification information for HN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-262: por_hni_unit_info

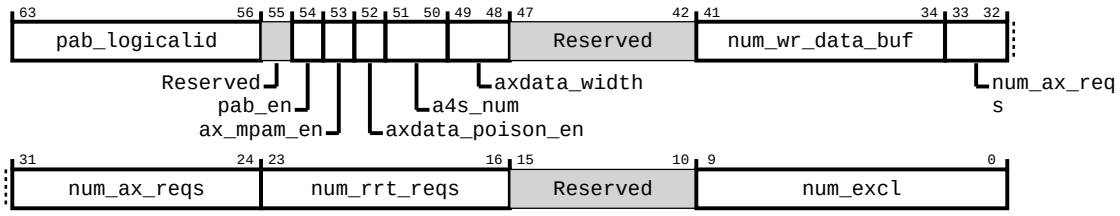


Table 4-278: por_hni_unit_info attributes

Bits	Name	Description	Type	Reset
[63:56]	pab_logicalid	PUB AUB bridge Logical ID	RO	Configuration dependent
[55]	Reserved	Reserved	RO	-
[54]	pab_en	PUB AUB bridge enable **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[53]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[52]	axdata_poison_en	Data poison support on ACE-Lite/AXI4 interface **1'b0** Not supported **1'b1** Supported	RO	Configuration dependent
[51:50]	a4s_num	Number of AXI4Stream interfaces present	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[49:48]	axdata_width	Data width on ACE-Lite/AXI4 interface **2'b00** 128 bits **2'b01** 256 bits **2'b10** 512 bits	RO	Configuration dependent
[47:42]	Reserved	Reserved	RO	-
[41:34]	num_wr_data_buf	Number of write data buffers in HN-I	RO	Configuration dependent
[33:24]	num_ax_reqs	Maximum number of outstanding ACE-Lite/AXI4 requests	RO	Configuration dependent
[23:16]	num_rrt_reqs	Number of CHI RRT request tracker entries in HN-I.	RO	Configuration dependent
[15:10]	Reserved	Reserved	RO	-
[9:0]	num_excl	Number of exclusive monitors in HN-I	RO	Configuration dependent

4.3.9.5 por_hni_unit_info_1

Provides component identification information for HN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-263: por_hni_unit_info_1

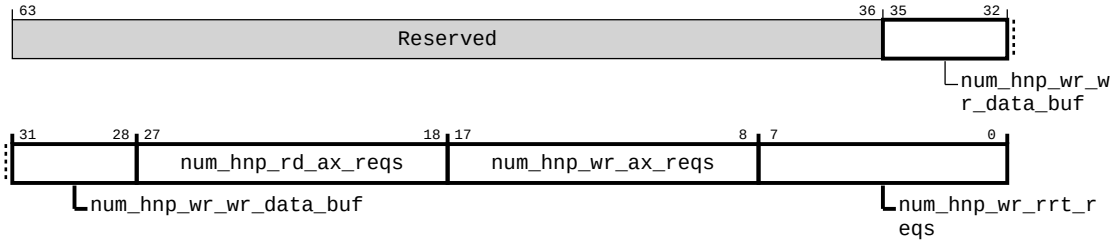


Table 4-279: por_hni_unit_info_1 attributes

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	-
[35:28]	num_hnp_wr_wr_data_buf	Number of P2P write data buffers in HN-I. HN-P only.	RO	Configuration dependent
[27:18]	num_hnp_rd_ax_reqs	Maximum number of outstanding P2P Read ACE-Lite/AXI4 requests. HN-P only.	RO	Configuration dependent
[17:8]	num_hnp_wr_ax_reqs	Maximum number of outstanding P2P Write ACE-Lite/AXI4 requests. HN-P only.	RO	Configuration dependent
[7:0]	num_hnp_wr_rrt_reqs	Number of P2P Write CHI RRT request tracker entries. HN-P only.	RO	Configuration dependent

4.3.9.6 por_hni_sam_addrregion0_cfg

Configures Address Region 0.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.sam

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-264: por_hni_sam_addrregion0_cfg

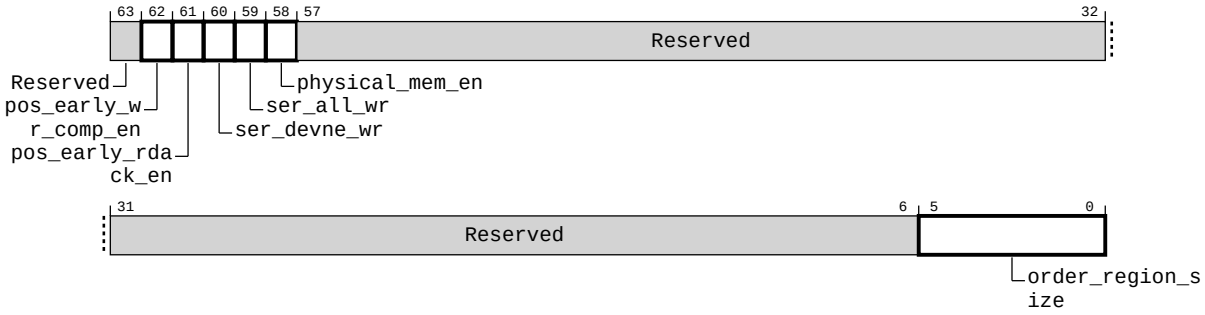


Table 4-280: por_hni_sam_addrregion0_cfg attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 0; used to improve write performance	RW	1'b1
[61]	pos_early_rdack_en	Enables sending early read receipts from HN-I in Address Region 0; used to improve ordered read performance	RW	1'b1
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 0	RW	1'b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 0	RW	1'b0
[58]	physical_mem_en	Address Region 0 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	1'b0
[57:6]	Reserved	Reserved	RO	-
[5:0]	order_region_size	<n>; used to calculate Order Region 0 size within Address Region 0 (2^n*4KB)	RW	6'b111111

4.3.9.7 por_hni_sam_addrregion1_cfg

Configures Address Region 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC08

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.sam

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-265: por_hni_sam_addrregion1_cfg

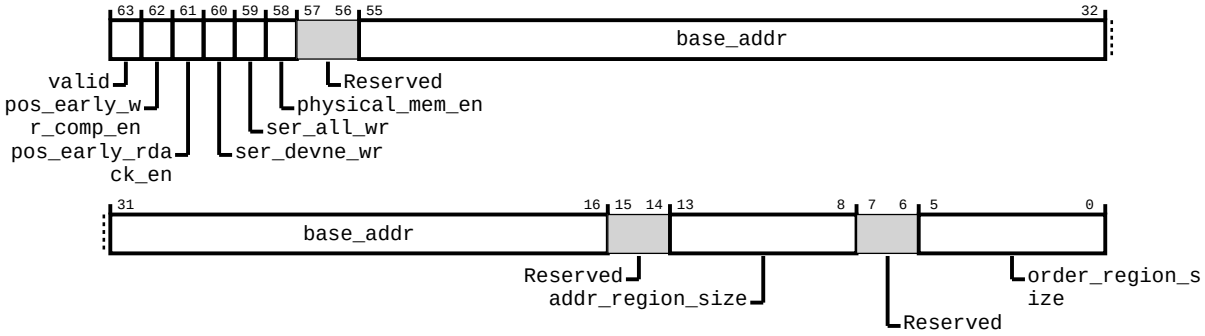


Table 4-281: por_hni_sam_addrregion1_cfg attributes

Bits	Name	Description	Type	Reset
[63]	valid	Address Region 1 fields are programmed and valid	RW	1'h0
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 1; used to improve write performance	RW	1'b1
[61]	pos_early_rdack_en	Enables sending early read receipts from HN-I in Address Region 1; used to improve ordered read performance	RW	1'b1
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 1	RW	1'b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 1	RW	1'b0
[58]	physical_mem_en	Address Region 1 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	1'b0
[57:56]	Reserved	Reserved	RO	-
[55:16]	base_addr	Address Region 1 base address; [address width-1:12] CONSTRAINT: Must be an integer multiple of the Address Region 1 size.	RW	40'h0
[15:14]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[13:8]	addr_region_size	<n>; used to calculate Address Region 1 size (2^n*4KB) CONSTRAINT: <n> must be configured so that the Address Region 1 size is less than or equal to 2^(address width).	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	order_region_size	<n>; used to calculate Order Region 1 size within Address Region 1 (2^n*4KB)	RW	6'h0

4.3.9.8 por_hni_sam_addrregion2_cfg

Configures Address Region 2.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC10`

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.sam

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-266: por_hni_sam_addrregion2_cfg

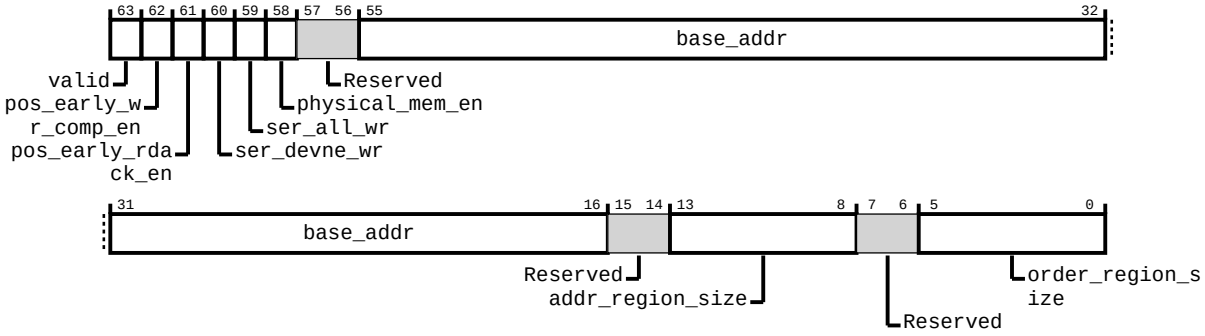


Table 4-282: por_hni_sam_addrregion2_cfg attributes

Bits	Name	Description	Type	Reset
[63]	valid	Address Region 2 fields are programmed and valid	RW	1'h0
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 2; used to improve write performance	RW	1'b1
[61]	pos_early_rdack_en	Enables sending early read receipts from HN-I in Address Region 2; used to improve ordered read performance	RW	1'b1
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 2	RW	1'b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 2	RW	1'b0
[58]	physical_mem_en	Address Region 2 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	1'b0
[57:56]	Reserved	Reserved	RO	-
[55:16]	base_addr	Address Region 2 base address; [address width-1:12] CONSTRAINT: Must be an integer multiple of the Address Region 2 size	RW	40'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	addr_region_size	<n>; used to calculate Address Region 2 size (2^n*4KB) CONSTRAINT: <n> must be configured so that the Address Region 2 size is less than or equal to 2^(address width).	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	order_region_size	<n>; used to calculate Order Region 2 size within Address Region 2 (2^n*4KB)	RW	6'h0

4.3.9.9 por_hni_sam_addrregion3_cfg

Configures Address Region 3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC18

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.sam

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-267: por_hni_sam_addrregion3_cfg

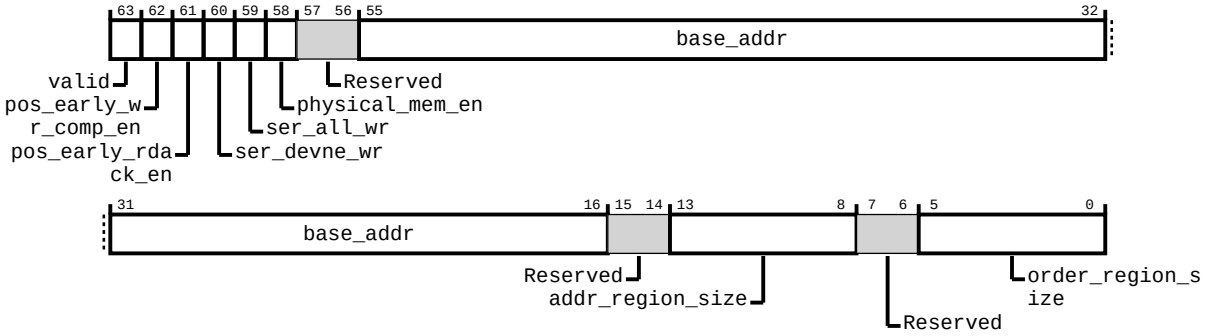


Table 4-283: por_hni_sam_addrregion3_cfg attributes

Bits	Name	Description	Type	Reset
[63]	valid	Fields of Address Region 3 are programmed and valid	RW	1'h0
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in Address Region 3; used to improve write performance	RW	1'b1
[61]	pos_early_rdack_en	Enables sending early read receipts from HN-I in Address Region 3; used to improve ordered read performance	RW	1'b1
[60]	ser_devne_wr	Used to serialize Device-nGnRnE writes within Address Region 3	RW	1'b0
[59]	ser_all_wr	Used to serialize all writes within Address Region 3	RW	1'b0
[58]	physical_mem_en	Address Region 3 follows Arm Architecture Reference Manual physical memory ordering guarantees	RW	1'b0
[57:56]	Reserved	Reserved	RO	-
[55:16]	base_addr	Address Region 3 base address; [address width-1:12] CONSTRAINT: Must be an integer multiple of the Address Region 3 size	RW	40'h0
[15:14]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[13:8]	addr_region_size	<n>; used to calculate Address Region 3 size (2^n*4KB) CONSTRAINT: <n> must be configured so that the Address Region 3 size is less than or equal to 2^(address width).	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	order_region_size	<n>; used to calculate Order Region 3 size within Address Region 3 (2^n*4KB)	RW	6'h0

4.3.9.10 por_hni_cfg_ctl

Functions as the configuration control register for HN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-268: por_hni_cfg_ctl

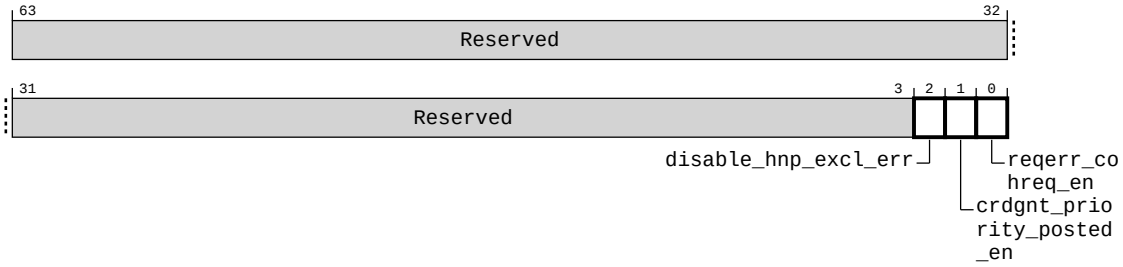


Table 4-284: por_hni_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	disable_hnp_excl_err	Disables sending NDE and Error logging on ReadNoSnp and WriteNoSnp Exclusives	RW	1'b0
[1]	crdgnt_priority_posted_en	Enables High priority Credit Grant responses to Posted requests	RW	1'b0
[0]	reqerr_cohreq_en	Enables sending of NDE response error to RN and logging of error information for the following requests: 1. Coherent Read 2. CleanUnique/MakeUnique 3. Coherent/CopyBack Write	RW	1'b1

4.3.9.11 por_hni_aux_ctl

Functions as the auxiliary control register for HN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-269: por_hni_aux_ctl

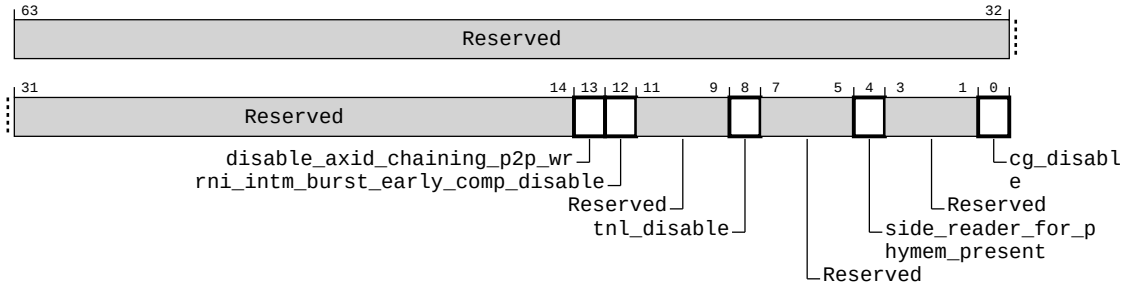


Table 4-285: por_hni_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13]	disable_axid_chaining_p2p_wr	Disables AXID based chaining of PCIe writes in P2P Write slice. HNP only	RW	1'b0
[12]	rni_intm_burst_early_comp_disable	Disables Early COMP to RNI for non-last burst writes	RW	1'b0
[11:9]	Reserved	Reserved	RO	-
[8]	tnl_disable	Disables RNI-HNI Tunneling in HNI. por_rni_aux_ctl.dis_hni_wr_stream must be set before setting this bit	RW	1'b0
[7:5]	Reserved	Reserved	RO	-
[4]	side_reader_for_phymem_present	Enables side reader in physical memory range	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	cg_disable	Disables HN-I architectural clock gates	RW	1'b0

4.3.9.12 por_hni_errfr

Functions as the error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3000`

Type

RO

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-270: por_hni_errfr

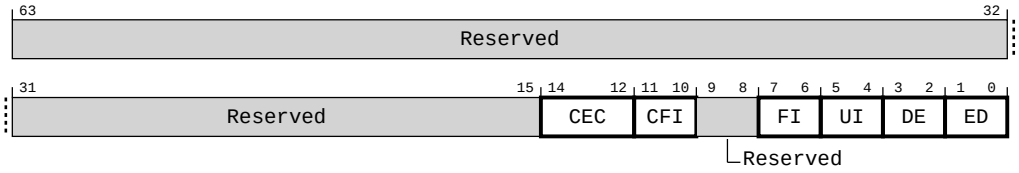


Table 4-286: por_hni_errfr attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism 3'b000: Does not implement standardized error counter model	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.9.13 por_hni_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3008

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-271: por_hni_errctlr

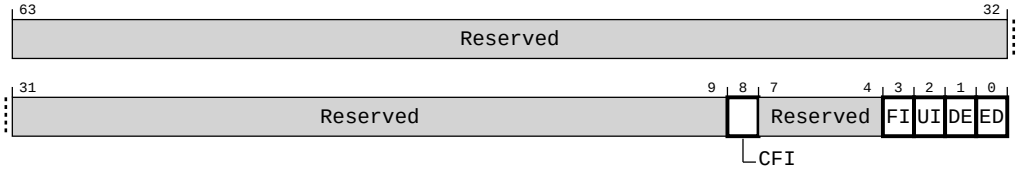


Table 4-287: por_hni_errctlr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_hni_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_hni_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_hni_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_hni_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_hni_errfr.ED	RW	1'b0

4.3.9.14 por_hni_errstatus

Functions as the error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3010`

Type

W1C

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-272: por_hni_errstatus

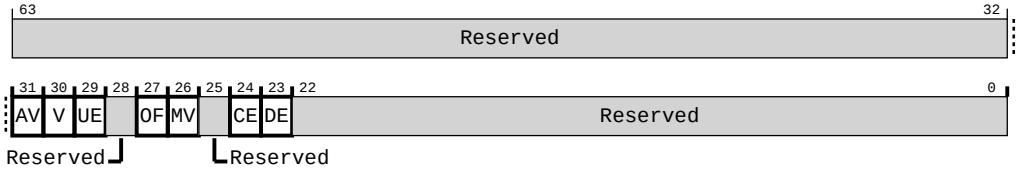


Table 4-288: por_hni_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; por_hni_erraddr contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	por_hni_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.9.15 por_hni_erraddr

Contains the error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3018`

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-273: por_hni_erraddr

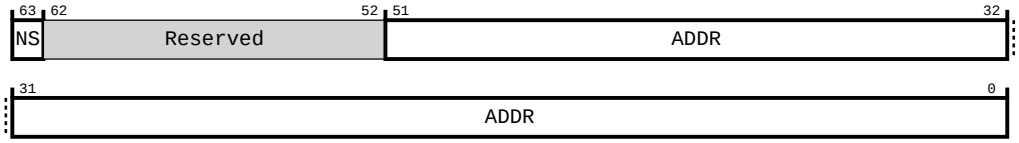


Table 4-289: por_hni_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: por_hni_erraddr.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.9.16 por_hni_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3020`

Type

RW

Reset value

See individual bit resets

Secure group override

por_hni_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-274: por_hni_errmisc

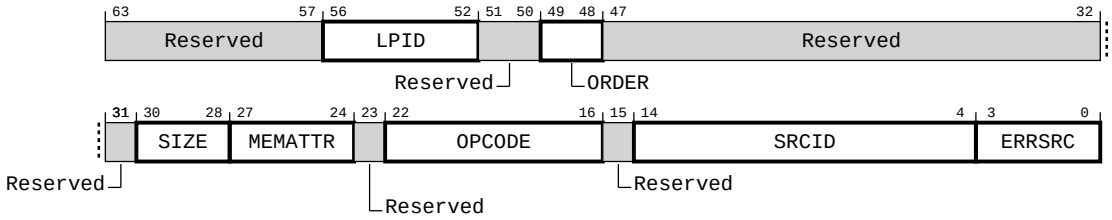


Table 4-290: por_hni_errmisc attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:52]	LPID	Error logic processor ID	RW	5'b0
[51:50]	Reserved	Reserved	RO	-
[49:48]	ORDER	Error order	RW	4'b0
[47:31]	Reserved	Reserved	RO	-
[30:28]	SIZE	Error transaction size	RW	3'b0
[27:24]	MEMATTR	Error memory attributes	RW	4'b0
[23]	Reserved	Reserved	RO	-
[22:16]	OPCODE	Error opcode	RW	7'b0
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	Error source **4'b0000** Coherent read **4'b0001** Coherent write **4'b0010** CleanUnique/MakeUnique **4'b0011** Atomic **4'b0100** Illegal configuration read **4'b0101** Illegal configuration write **4'b0110** Configuration write data partial byte enable error **4'b0111** Configuration write data parity error or poison error **4'b1000** BRESP error **4'b1001** Poison error **4'b1010** BRESP error and poison error **4'b1011** Unsupported Exclusive access (HN-P only) NOTE: For configuration write data, BRESP, and poison errors, por_hni_errmisc.SRCID is the only valid field. For other error types, all fields are valid.	RW	4'b0

4.3.9.17 por_hni_errfr_NS

Functions as the Non-secure error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3100`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-275: por_hni_errfr_NS

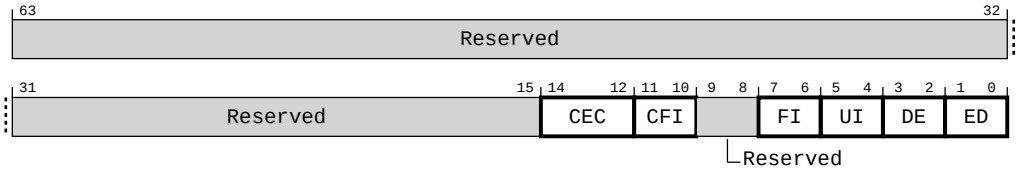


Table 4-291: por_hni_errfr_NS attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism 3'b000: Does not implement standardized error counter model	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.9.18 por_hni_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3108`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-276: por_hni_errctlr_NS

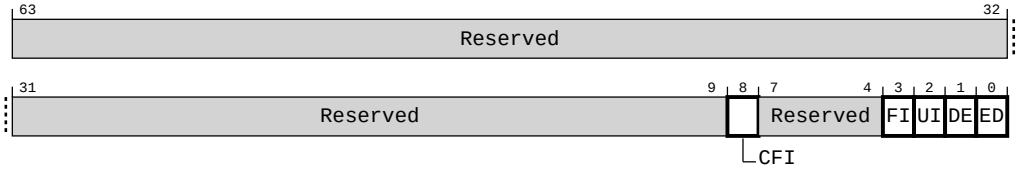


Table 4-292: por_hni_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_hni_errfr_NS.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_hni_errfr_NS.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_hni_errfr_NS.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_hni_errfr_NS.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_hni_errfr_NS.ED	RW	1'b0

4.3.9.19 por_hni_errstatus_NS

Functions as the Non-secure error status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3110`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-277: por_hni_errstatus_NS

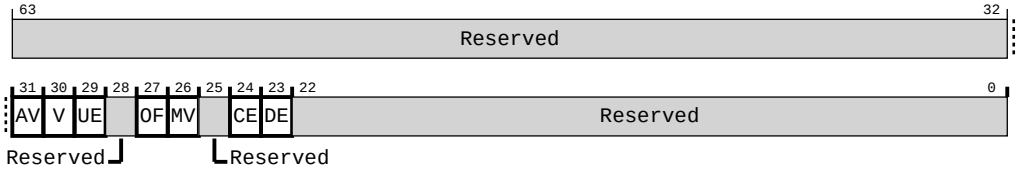


Table 4-293: por_hni_errstatus_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; por_hni_erraddr_NS contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	por_hni_errmisc_NS valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.9.20 por_hni_erraddr_NS

Contains the Non-secure error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3118`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-278: por_hni_erraddr_NS

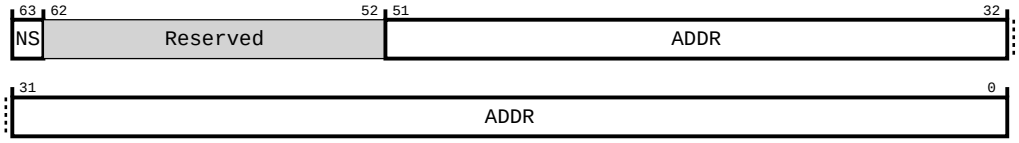


Table 4-294: por_hni_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: por_hni_erraddr_NS.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.9.21 por_hni_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3120`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-279: por_hni_errmisc_NS

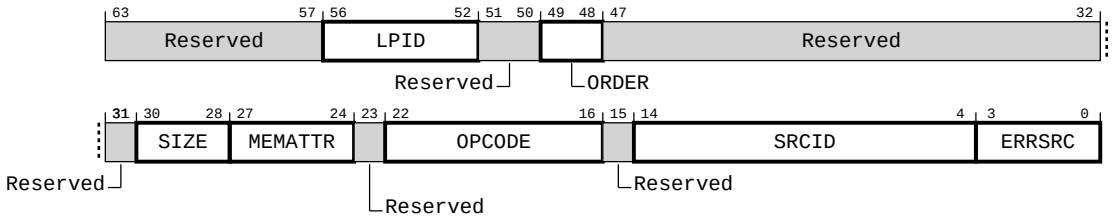


Table 4-295: por_hni_errmisc_NS attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:52]	LPID	Error logic processor ID	RW	5'b0
[51:50]	Reserved	Reserved	RO	-
[49:48]	ORDER	Error order	RW	4'b0
[47:31]	Reserved	Reserved	RO	-
[30:28]	SIZE	Error transaction size	RW	3'b0
[27:24]	MEMATTR	Error memory attributes	RW	4'b0
[23]	Reserved	Reserved	RO	-
[22:16]	OPCODE	Error opcode	RW	7'b0
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0
[3:0]	ERRSRC	Error source **4'b0000** Coherent read **4'b0001** Coherent write **4'b0010** CleanUnique/MakeUnique **4'b0011** Atomic **4'b0100** Illegal configuration read **4'b0101** Illegal configuration write **4'b0110** Configuration write data partial byte enable error **4'b0111** Configuration write data parity error or poison error **4'b1000** BRESP error **4'b1001** Poison error **4'b1010** BRESP error and poison error **4'b1011** Unsupported Exclusive access (HN-P only) NOTE: For configuration write data, BRESP, and poison errors, por_hni_errmisc_NS.SRCID is the only valid field. For other error types, all fields are valid.	RW	4'b0

4.3.9.22 por_hni_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-280: por_hni_pmu_event_sel

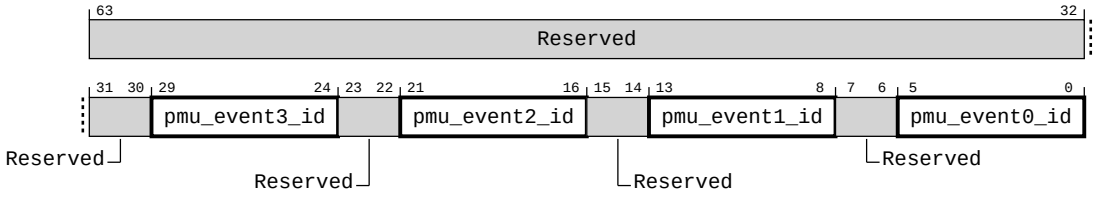


Table 4-296: por_hni_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	HN-I PMU Event 3 select; see pmu_event0_id for encodings	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	HN-I PMU Event 2 select; see pmu_event0_id for encodings	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	HN-I PMU Event 1 select; see pmu_event0_id for encodings	RW	6'b0
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	HN-I PMU Event 0 select **6'h00** No event **6'h20** RRT read occupancy count overflow **6'h21** RRT write occupancy count overflow **6'h22** RDT read occupancy count overflow **6'h23** RDT write occupancy count overflow **6'h24** WDB occupancy count overflow **6'h25** RRT read allocation **6'h26** RRT write allocation **6'h27** RDT read allocation **6'h28** RDT write allocation **6'h29** WDB allocation **6'h2A** RETRYACK TXRSP flit sent **6'h2B** ARVALID set without ARREADY event **6'h2C** ARREADY set without ARVALID event **6'h2D** AWVALID set without AWREADY event **6'h2E** AWREADY set without AWVALID event **6'h2F** WVALID set without WREADY event **6'h30** TXDAT stall (TXDAT valid but no link credit available) **6'h31** Non-PCIe serialization event **6'h32** PCIe serialization event NOTE: All other encodings are reserved.	RW	6'b0

4.3.9.23 por_hnp_pmu_event_sel

Specifies the PMU event to be counted. HNP only

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2008`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-281: por_hnp_pmu_event_sel

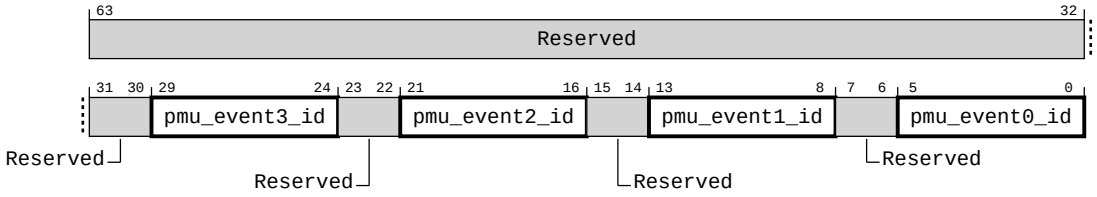


Table 4-297: por_hnp_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	P2P Slice PMU Event 3 select; see pmu_event0_id for encodings"	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	P2P Slice PMU Event 2 select; see pmu_event0_id for encodings	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	P2P Slice PMU Event 1 select; see pmu_event0_id for encodings	RW	6'b0
[7:6]	Reserved	Reserved	RO	-
[5:0]	pmu_event0_id	P2P Slice PMU Event 0 select **6'h00** No event **6'h01** RRT write occupancy count overflow **6'h02** RDT write occupancy count overflow **6'h03** WDB occupancy count overflow **6'h04** RRT write allocation **6'h05** RDT write allocation **6'h06** WDB allocation **6'h07** without AWREADY event **6'h08** AWREADY set without AWVALID event **6'h09** WVALID set without WREADY event **6'h11** RRT read occupancy count overflow **6'h12** RDT read occupancy count overflow **6'h13** RRT read allocation **6'h14** RDT read allocation **6'h15** ARVALID set without ARREADY event **6'h16** ARREADY set without ARVALID event NOTE: All other encodings are reserved.	RW	6'b0

4.3.10 HN-F register descriptions

This section lists the HN-F registers.

4.3.10.1 cmn_hns_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-282: cmn_hns_node_info

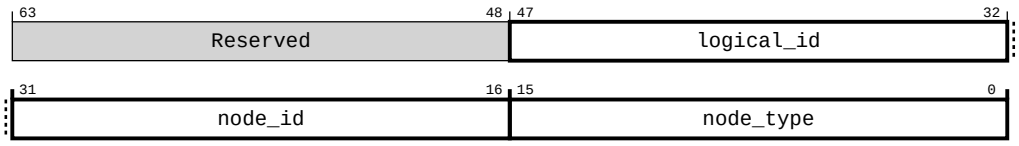


Table 4-298: cmn_hns_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	$logical_id_description	RO	Configuration dependent
[31:16]	node_id	$node_id_description	RO	Configuration dependent
[15:0]	node_type	$node_type_description	RO	Configuration dependent

4.3.10.2 cmn_hns_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-283: cmn_hns_child_info

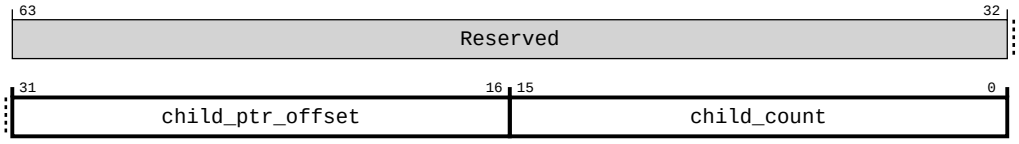


Table 4-299: cmn_hns_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.10.3 cmn_hns_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-284: cmn_hns_secure_register_groups_override

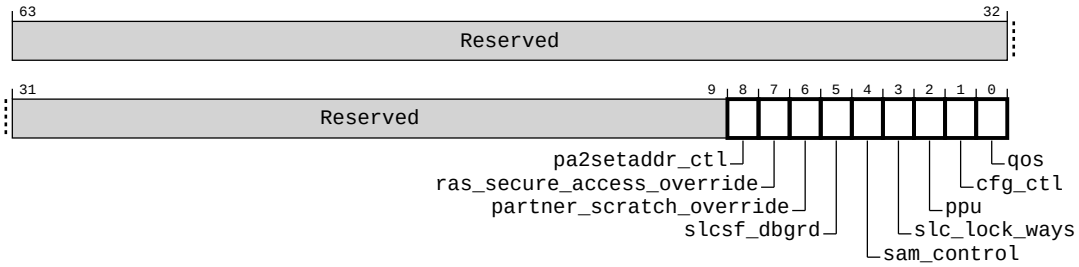


Table 4-300: cmn_hns_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	pa2setaddr_ctl	Allow Non-secure access to Secure PA2SETADDR registers	RW	1'b0
[7]	ras_secure_access_override	Allow Non-secure access to Secure RAS registers	RW	1'b0
[6]	partner_scratch_override	Allows Non-secure access to Secure Partner scratch registers	RW	1'b0
[5]	slcsf_dbgrd	Allows Non-secure access to Secure SLC/SF debug read registers	RW	1'b0
[4]	sam_control	Allows Non-secure access to Secure HN-F SAM control registers	RW	1'b0
[3]	slc_lock_ways	Allows Non-secure access to Secure cache way locking registers	RW	1'b0
[2]	ppu	Allows Non-secure access to Secure power policy registers	RW	1'b0
[1]	cfg_ctl	Allows Non-secure access to Secure configuration control register (cmn_hns_cfg_ctl)	RW	1'b0
[0]	qos	Allows Non-secure access to Secure QoS registers	RW	1'b0

4.3.10.4 cmn_hns_unit_info

Provides component identification information for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-285: cmn_hns_unit_info

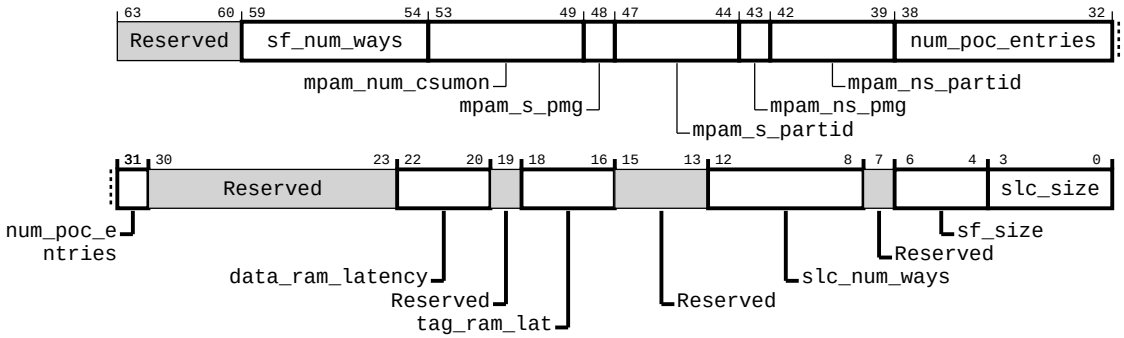


Table 4-301: cmn_hns_unit_info attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:54]	sf_num_ways	Number of cache ways in the SF	RO	-
[53:49]	mpam_num_csumon	Number of Cache Storage Usage Monitors for MPAM	RO	Configuration dependent
[48]	mpam_s_pmg	MPAM Secure supported PMGs **1'b0** 1 PMG **1'b1** 2 S PMG	RO	-

Bits	Name	Description	Type	Reset
[47:44]	mpam_s_partid	MPAM Secure supported PARTIDs **4'b0000** 1 S PARTID **4'b0001** 2 S PARTID **4'b0010** 4 S PARTID **4'b0011** 8 S PARTID **4'b0100** 16 S PARTID **4'b0101** 32 S PARTID **4'b0110** 64 S PARTID **4'b0111** 128 S PARTID **4'b1000** 256 S PARTID **4'b1001** 512 S PARTID	RO	-
[43]	mpam_ns_pmg	MPAM Non-secure supported PMGs **1'b0** 1 NS PMG **1'b1** 2 NS PMG	RO	-
[42:39]	mpam_ns_partid	MPAM Non-secure supported PARTIDs **4'b0000** 1 NS PARTID **4'b0001** 2 NS PARTID **4'b0010** 4 NS PARTID **4'b0011** 8 NS PARTID **4'b0100** 16 NS PARTID **4'b0101** 32 NS PARTID **4'b0110** 64 NS PARTID **4'b0111** 128 NS PARTID **4'b1000** 256 NS PARTID **4'b1001** 512 NS PARTID	RO	-
[38:31]	num_poc_entries	Number of POCQ entries	RO	Configuration dependent
[30:23]	Reserved	Reserved	RO	-
[22:20]	data_ram_latency	SLC data RAM latency (in cycles)	RO	-
[19]	Reserved	Reserved	RO	-
[18:16]	tag_ram_lat	SLC tag RAM latency (in cycles)	RO	-
[15:13]	Reserved	Reserved	RO	-
[12:8]	slc_num_ways	Number of cache ways in the SLC	RO	-
[7]	Reserved	Reserved	RO	-
[6:4]	sf_size	SF size **3'b000** (32KB * sf_num_ways) **3'b001** (64KB * sf_num_ways) **3'b010** (128KB * sf_num_ways) **3'b011** (256KB * sf_num_ways) **3'b101** (512KB * sf_num_ways)	RO	-

Bits	Name	Description	Type	Reset
[3:0]	slc_size	SLC size **4'b0000** No SLC **4'b0001** 128KB **4'b0010** 256KB **4'b0011** 512KB **4'b0100** 1MB **4'b0101** 1.5MB **4'b0110** 2MB **4'b0111** 3MB **4'b1000** 4MB **4'b1001** 384KB	RO	-

4.3.10.5 cmn_hns_unit_info_1

Provides component identification information for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-286: cmn_hns_unit_info_1

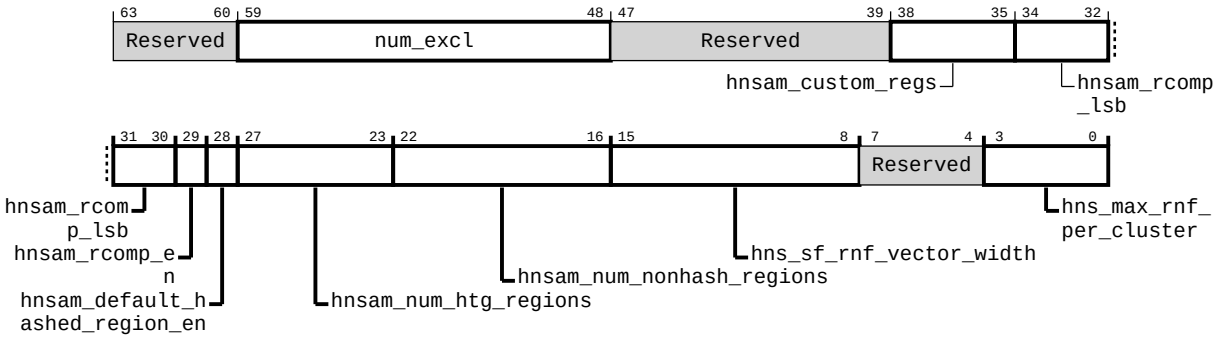


Table 4-302: cmn_hns_unit_info_1 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:48]	num_excl	Number of exclusive monitors	RO	-
[47:39]	Reserved	Reserved	RO	-
[38:35]	hnsam_custom_regs	Number of customer specific registers for customer implemented logic	RO	Configuration dependent
[34:30]	hnsam_rcomp_lsb	Defines the minimum size of HTG when POR_HNSAM_RCOMP_EN_PARAM = 1, 20 value defines minimum size as 1MB and 26 value defines minimum size as 64MB	RO	Configuration dependent
[29]	hnsam_rcomp_en	Enable Range based address comparison for HNSAM HTG/Nonhashed groups. Program start address and end address	RO	Configuration dependent
[28]	hnsam_default_hashed_region_en	Enable default hashed group for HNSAM. To support backward compatible, set this parameter	RO	Configuration dependent
[27:23]	hnsam_num_htg_regions	Number of HTG regions supported by the HNSAM	RO	Configuration dependent
[22:16]	hnsam_num_nonhash_regions	Number of non-hashed regions supported by the HNSAM	RO	Configuration dependent
[15:8]	hns_sf_rnf_vector_width	Total Number of bits in RNF tracking vector in the Snoop Filter (Total SF_VEC_WIDTH = (TOTAL_RNF/ HNS_MAX_CLUSTER_PARAM)+HNS_SF_ADD_VECTOR_WIDTH)	RO	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3:0]	hns_max_rnf_per_cluster	Describes the maximum number of RN-F's in a single cluster	RO	Configuration dependent

4.3.10.6 cmn_hns_cfg_ctl

Functions as the configuration control register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA00

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-287: cmn_hns_cfg_ctl

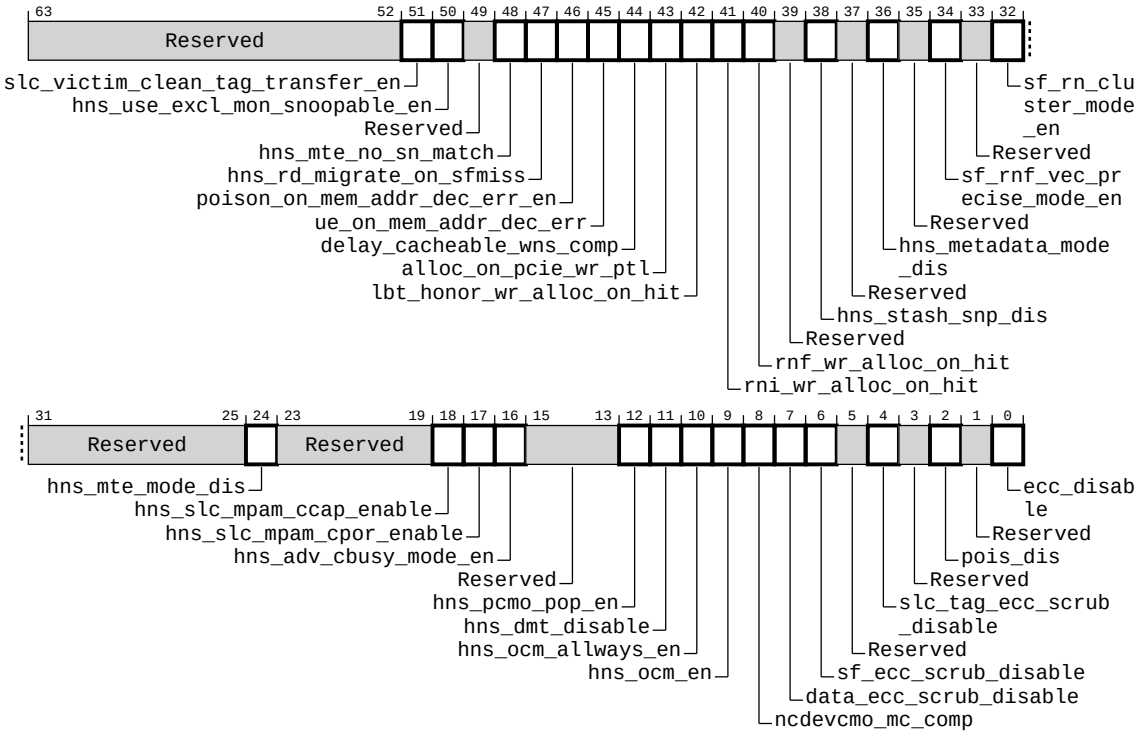


Table 4-303: cmn_hns_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	slc_victim_clean_tag_transfer_en	When set, HNS propagates clean tag to SN when SLC victim has clean tag	RW	1'b0
[50]	hns_use_excl_mon_snoopable_en	When set, HNS uses exclusive monitor for snoopable traffic in imprecise SF modes	RW	1'b0
[49]	Reserved	Reserved	RO	-
[48]	hns_mte_no_sn_match	When set, HNS does MTE match locally without propagating to SN	RW	1'b0
[47]	hns_rd_migrate_on_sfmiss	Migrates a read from LCC/SLC if sf miss	RW	1'b1
[46]	poison_on_mem_addr_dec_err_en	When set, set poison in read data for CXL address decode error	RW	1'b1
[45]	ue_on_mem_addr_dec_err	Log CXL address decode error as UE in error register	RW	1'b0
[44]	delay_cacheable_wns_comp	Sends late completion for cacheable WriteNoSnoop	RW	1'b0
[43]	alloc_on_pcie_wr_ptl	Forces HBT PCIE partial writes to allocate in SLC	RW	1'b0
[42]	lbt_honor_wr_alloc_on_hit	Forces LBT Write requests to honor wlu_alloc_on_hit, rnf_wr_alloc_on_hit, and rni_wr_alloc_on_hit	RW	1'b0
[41]	rni_wr_alloc_on_hit	Forces RNI Write requests to allocate if the line hit in SLC	RW	1'b0
[40]	rnf_wr_alloc_on_hit	Forces RNF Write requests to allocate if the line hit in SLC	RW	1'b0
[39]	Reserved	Reserved	RO	-
[38]	hns_stash_snp_dis	Disables stashing snoop in HNS when set to 1'b1	RW	1'b0
[37]	Reserved	Reserved	RO	-
[36]	hns_metadata_mode_dis	Disables the METADATA features in HNS when set to 1'b1	RW	1'b0
[35]	Reserved	Reserved	RO	-
[34]	sf_rnf_vec_precise_mode_en	Enables the snoop filter's precise RNF vector in clustered mode when set to 1'b1	RW	1'b1
[33]	Reserved	Reserved	RO	-
[32]	sf_rn_cluster_mode_en	Enables the snoop filter clustering of the RN-F ID's using programmable registers	RW	1'b1
[31:25]	Reserved	Reserved	RO	-
[24]	hns_mte_mode_dis	Disables the MTE features in HNS when set to 1'b1	RW	1'b0
[23:19]	Reserved	Reserved	RO	-
[18]	hns_slc_mpam_ccap_enable	Enable MPAM Cache Capacity Partitioning for SLC **1'b1** Cache Capacity Partitioning is enabled if supported in Hardware. **1'b0** Cache Capacity Partitioning is disabled for SLC. NOTE: If MPAM is disabled at build time, this bit has no meaning.	RW	1'b0

Bits	Name	Description	Type	Reset
[17]	hns_slc_mpam_cpor_enable	Enable MPAM Cache Portion Partitioning for SLC **1'b1** Cache Portion Partitioning is enabled if supported in Hardware. **1'b0** Cache Portion Partitioning is disabled for SLC. . NOTE: If MPAM is disabled at build time, this bit has no meaning.	RW	1'b0
[16]	hns_adv_cbusy_mode_en	Enables the advanced features of HNS CBusy handling	RW	1'b0
[15:13]	Reserved	Reserved	RO	-
[12]	hns_pcmo_pop_en	Terminates PCMO in HNS when this bit is set to 1'b1	RW	1'b0
[11]	hns_dmt_disable	Disables DMT when set	RW	1'b0
[10]	hns_ocm_allways_en	Enables all SLC ways with OCM	RW	1'b0
[9]	hns_ocm_en	Enables region locking with OCM support	RW	1'b0
[8]	ncdevcmo_mc_comp	Disables HN-F completion when set NOTE: When set, HN-F sends completion for the following transactions received after completion from SN: 1. Non-Cacheable WriteNoSnp 2. Device WriteNoSnp 3. CMO (cache maintenance operations) CONSTRAINT: When this bit is set, por_rni_cfg_ctl.dis_ncwr_stream and por_rnd_cfg_ctl.dis_ncwr_stream must also be set.	RW	1'b0
[7]	data_ecc_scrub_disable	Disables data single-bit ECC error scrubbing for non-migrating reads when set	RW	1'b1
[6]	sf_ecc_scrub_disable	Disables SF tag single-bit ECC error scrubbing when set	RW	1'b0
[5]	Reserved	Reserved	RO	-
[4]	slc_tag_ecc_scrub_disable	Disables SLC tag single-bit ECC error scrubbing when set	RW	1'b0
[3]	Reserved	Reserved	RO	-
[2]	pois_dis	Disables parity error data poison when set	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	ecc_disable	Disables SLC and SF ECC generation/detection when set	RW	1'b0

4.3.10.7 cmn_hns_aux_ctl

Functions as the auxiliary control register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-288: cmn_hns_aux_ctl

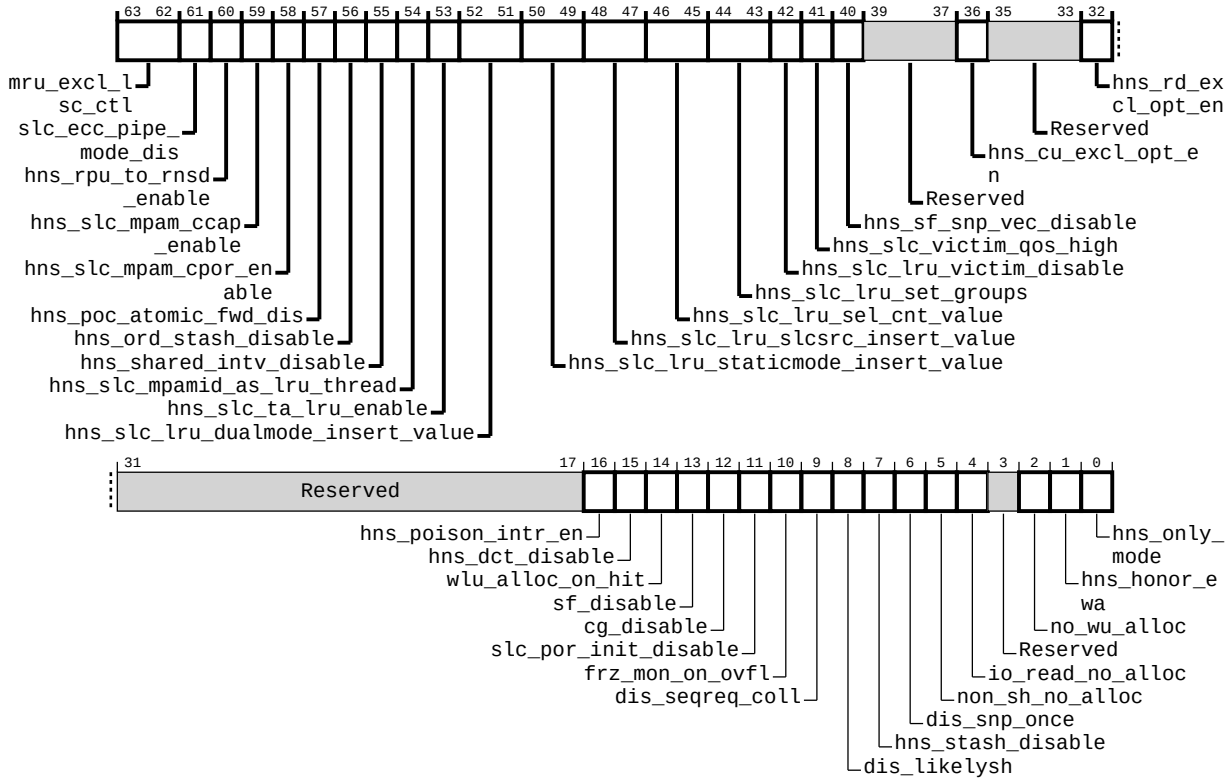


Table 4-304: cmn_hns_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:62]	mru_excl_lsc_ctl	MRU Exclusive control for LSC.	RW	2'b00
[61]	slc_ecc_pipe_mode_dis	Disables inline ECC pipe mode in SLC. CONSTRAINT: Must be programmed at boot time.	RW	1'b1
[60]	hns_rpu_to_rnsd_enable	Enables HN-F to treat ReadPrefUnique ops as ReadNotSharedDirty	RW	1'b0

Bits	Name	Description	Type	Reset
[59]	hns_slc_mpam_ccap_enable	Enable MPAM Cache Capacity Partitioning for SLC **1'b1** Cache Capacity Partitioning is enabled if supported in Hardware. **1'b0** Cache Capacity Partitioning is disabled for SLC. NOTE: This bit moved to cfg_ctl and will be removed in future version of ICN. NOTE: If MPAM is disabled at build time, this bit has no meaning.	RW	1'b0
[58]	hns_slc_mpam_cpor_enable	Enable MPAM Cache Portion Partitioning for SLC **1'b1** Cache Portion Partitioning is enabled if supported in Hardware. **1'b0** Cache Portion Partitioning is disabled for SLC. . NOTE: This bit moved to cfg_ctl and will be removed in future version of ICN. NOTE: If MPAM is disabled at build time, this bit has no meaning.	RW	1'b0
[57]	hns_poc_atomic_fwd_dis	Disable the atomic data forwarding in POCQ	RW	1'b0
[56]	hns_ord_stash_disable	Disables stash operation for ordered write stash requests	RW	1'b0
[55]	hns_shared_intv_disable	Disables snoop requests to CHIB RN-F with shared copy	RW	Configuration dependent
[54]	hns_slc_mpamid_as_lru_thread	Use MPAM PARTID as ThreadID for Thread Aware eLRU **1'b1** ThreadID is based on MPAM PARTID+NS for Thread Aware eLRU. **1'b0** ThreadID is based on LPID+LID for Thread Aware eLRU. Note: MPAM PARTID is used only if MPAM is enabled.	RW	1'b0
[53]	hns_slc_ta_lru_enable	Thread Aware eLRU enable **1'b0** ThreadID used for eLRU is zero. **1'b1** ThreadID used for eLRU is based on MPAMID or LPID+LID. Note: If SLC size is less than 256KB, this bit is ignore.	RW	1'b0
[52:51]	hns_slc_lru_dualmode_insert_value	Insertion value for Dual mode eLRU NOTE: Default is 2'b11.	RW	2'b11
[50:49]	hns_slc_lru_staticmode_insert_value	Insertion value for Static mode eLRU NOTE: Default is 2'b10.	RW	2'b10
[48:47]	hns_slc_lru_slcsrc_insert_value	Insertion value if SLC source bit is set NOTE: Default is 2'b00.	RW	2'b00

Bits	Name	Description	Type	Reset
[46:45]	hns_slc_lru_sel_cnt_value	Selection counter value for eLRU to determine which group policy is more effective **2'b00** Sel counter is like an 8-bit range; upper limit is 255; middle point is 128 **2'b01** Sel counter is like a 9-bit range; upper limit is 511; middle point is 256 **2'b10** Sel counter is like a 10-bit range; upper limit is 1023; middle point is 512 **2'b11** Sel counter is like an 11-bit range; upper limit is 2047; middle point is 1024	RW	2'b10
[44:43]	hns_slc_lru_set_groups	Number of sets in monitor group for enhance LRU **2'b00** 16 **2'b01** 32 **2'b10** 64 **2'b11** 128 NOTE: Default is 32 sets per monitor group. If cache size is small (128KB or less), there would be only one set per group.	RW	2'b01
[42]	hns_slc_lru_victim_disable	Disable enhanced LRU based victim selection for SLC **1'b0** SLC victim selection is based on eLRU. **1'b1** SLC victim selection is based on LFSR. NOTE: Victim selection for SF is always LFSR-based.	RW	1'b1
[41]	hns_slc_victim_qos_high	SLC victim QoS behavior for SN write request **1'b0** Each victim inherits the QoS value of the request which caused it **1'b1** All victims use high QoS class (14)	RW	1'b0
[40]	hns_sf_snp_vec_disable	Disables SF snoop vector when set	RW	1'b0
[39:37]	Reserved	Reserved	RO	-
[36]	hns_cu_excl_opt_en	CleanUnique exclusive optimization enable	RW	1'b1
[35:33]	Reserved	Reserved	RO	-
[32]	hns_rd_excl_opt_en	ReadNotSharedDirty exclusive optimization enable	RW	1'b0
[31:17]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[16]	hns_poison_intr_en	Enables reporting an interrupt by HN-F when poison is detected at SLC	RW	Configuration dependent
[15]	hns_dct_disable	Disables DCT when set	RW	Configuration dependent
[14]	wlu_alloc_on_hit	Forces WLU requests to allocate if the line hit in SLC	RW	1'b0
[13]	sf_disable	Disables SF	RW	1'b0
[12]	cg_disable	Disables HN-F architectural clock gates	RW	1'b0
[11]	slc_por_init_disable	Disables SLC and SF initialization on Reset	RW	1'b0
[10]	frz_mon_on_ovfl	Freezes the exclusive monitors	RW	1'b0
[9]	dis_seqreq_coll	-	RW	1'b0
[8]	dis_likelysh	Disables Likely Shared based allocations	RW	1'b0
[7]	hns_stash_disable	Disables HN-F stash support	RW	Configuration dependent
[6]	dis_snp_once	When set, disables SnpOnce and converts to SnpShared	RW	Configuration dependent
[5]	non_sh_no_alloc	Disables SLC allocation for non-shareable cacheable transactions when set	RW	1'b0
[4]	io_read_no_alloc	When set, disables ReadOnce and ReadNoSnp allocation in SLC from RN-Is	RW	1'b0
[3]	Reserved	Reserved	RO	-
[2]	no_wu_alloc	Disables WriteUnique/WriteLineUnique allocations in SLC when set	RW	1'b0
[1]	hns_honor_ewa	When set, postpones completion for writes where EWA=0 in the request until HN-F receives completion from MC or SBSX	RW	1'b1
[0]	hns_only_mode	Enables HN-F only mode; disables SLC and SF when set	RW	1'b0

4.3.10.8 cmn_hns_aux_ctl_1

Functions as the auxiliary control register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA10`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-289: cmn_hns_aux_ctl_1

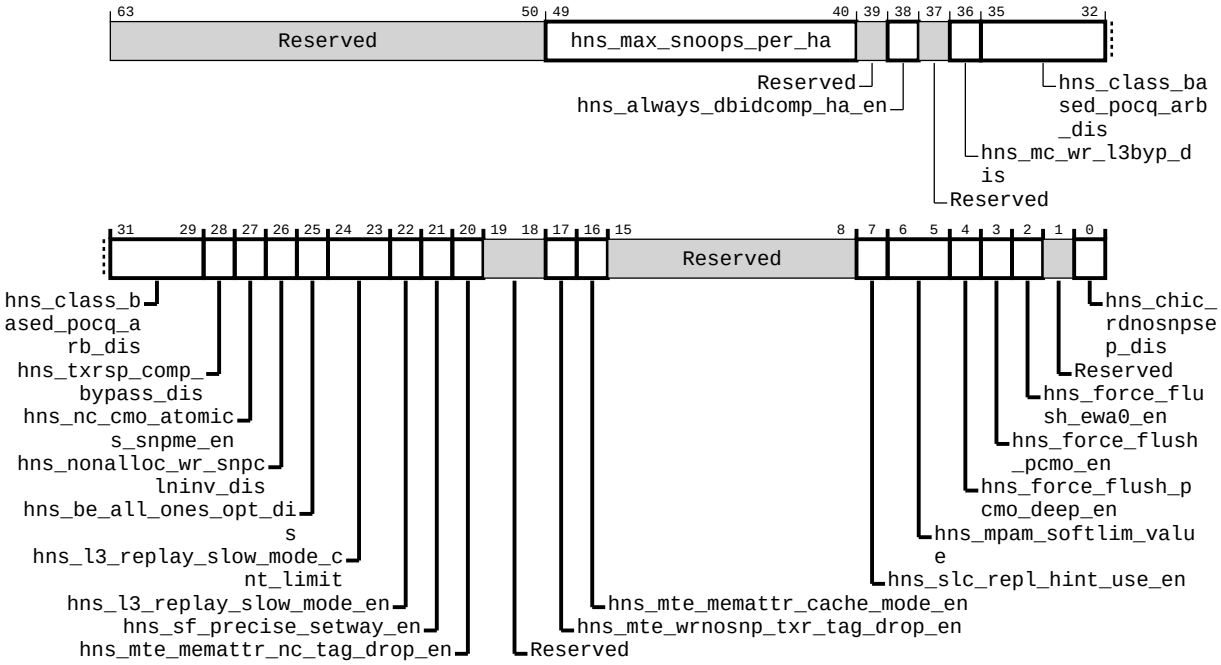


Table 4-305: cmn_hns_aux_ctl_1 attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:40]	hns_max_snoops_per_ha	HNS will use the register value instead of the parameter to control Max snoops per HA	RW	Configuration dependent
[39]	Reserved	Reserved	RO	-
[38]	hns_always_dbidcomp_ha_en	When set, HNS will combine DBID and Comp response for all writeunique requests from CXHA to post SLC/SF lookup and snoops	RW	1'b0
[37]	Reserved	Reserved	RO	-
[36]	hns_mc_wr_l3byp_dis	When set, disables l3 bypass path to mc request for writes	RW	1'b0

Bits	Name	Description	Type	Reset
[35:29]	hns_class_based_pocq_arb_dis	Disables Class based arbitration for various POCQ arbiters. For each bit: **1'b0** Use Class based arbitration **1'b1** Use QoS based arbitration Legacy mode. **[35]** POCQ entry selection for SN Static Credit Grant return. **[34]** POCQ entry selection for SLC/SF pipeline request **[33]** POCQ entry selection for TXRSP. **[32]** POCQ entry selection for TXDAT. **[31]** POCQ entry selection for ADQ **[29]** Reserved for future use	RW	7'b0000000
[28]	hns_txrsp_comp_bypass_dis	When set, TXRSP COMP bypass gets disabled for WEOE/EVICT	RW	1'b1
[27]	hns_nc_cmo_atomics_snpme_en	When set to 1, all incoming non-cachable atomics and cmo's from RNF will be back-snooped	RW	1'b0
[26]	hns_nonalloc_wr_snpclninv_dis	Disable the snp type of snp_cln_inv on non-allocating writes. Send snp_uniq instead	RW	1'b0
[25]	hns_be_all_ones_opt_dis	Disable the optimizations related to BE=1's hint on WR_PTL from RNI	RW	1'b0
[24:23]	hns_l3_replay_slow_mode_cnt_limit	L3 arbitration throttle count limit, when enabled. **00** L3 Throttle is enabled after 512 setway haz replays **01** L3 Throttle is enabled after 1024 setway haz replays **10** L3 Throttle is enabled after 2048 setway haz replays **11** L3 Throttle is enabled after 4096 setway haz replays	RW	2'b00
[22]	hns_l3_replay_slow_mode_en	Enables L3 arbitration slow mode in case of constant replays, when set to 1'b1	RW	1'b0
[21]	hns_sf_precise_setway_en	Enables Precise setway hazard, when set to 1'b1	RW	1'b0
[20]	hns_mte_memattr_nc_tag_drop_en	Enables HNS to drop any dirty tags for Non-Cacheable memory, when set to 1'b1	RW	1'b0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_mte_wrnosnp_txr_tag_drop_en	When set to 1'b1, HNS will drop clean tags from a WriteNoSnp with tagop Transfer	RW	1'b0

Bits	Name	Description	Type	Reset
[16]	hns_mte_memattr_cache_mode_en	When set to 1'b1, it enables HNS to convert Non-cacheable requests to cacheable if MTE tags are required	RW	1'b1
[15:8]	Reserved	Reserved	RO	-
[7]	hns_slc_repl_hint_use_en	**1'b0** Interconnect generated SLC Replacement hints are used for eLRU. **1'b1** RN-F provided SLC Replacement hints are used for eLRU.	RW	1'b1
[6:5]	hns_mpam_softlim_value	**2'b00** Soft limit is 0% below hardlimit **2'b01** Soft limit is 3.13% (1/32) below hardlimit **2'b10** Soft limit is 6.25% (1/16) below hardlimit **2'b11** Soft limit is 9.38% (3/32) below hardlimit NOTE: Default is 3.13% below hardlimit. If CMAX value set is at or below 12.5%, soft limit is ignored.	RW	2'b01
[4]	hns_force_flush_pcmo_deep_en	Make PCMO request for SLC and SF flush generated SN writes as Deep PCMO. CONSTRAINT: hns_force_flush_pcmo_deep_en is valid only if hns_force_flush_pcmo_en bit is set. CONSTRAINT: This bit can be set only if ALL SNs in the system support deep attribute.	RW	1'b0
[3]	hns_force_flush_pcmo_en	Generate PCMO request for SLC and SF flush generated SN writes	RW	1'b0
[2]	hns_force_flush_ewa0_en	Force SLC and SF flush to use EWA 0 for SN writes	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	hns_chic_rdnosnpsep_dis	Disables separation of Data and Comp in CHIC mode	RW	1'b0

4.3.10.9 cmn_hns_cbusy_limit_ctl

Cbusy threshold limits for POCQ entries.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA18`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-290: cmn_hns_cbusy_limit_ctl

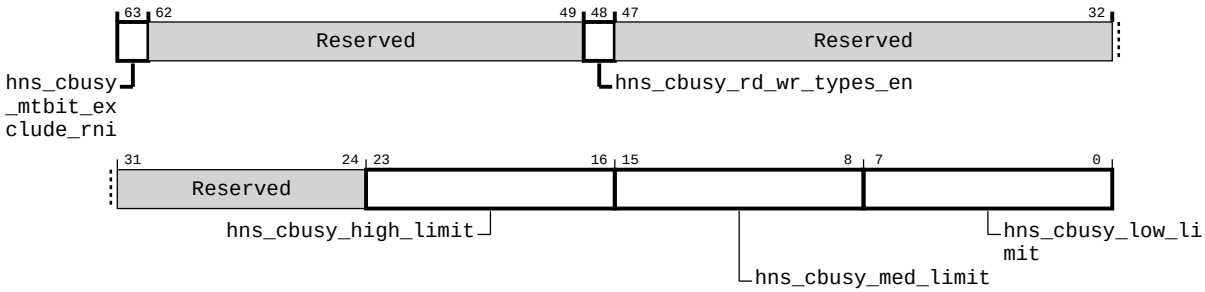


Table 4-306: cmn_hns_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63]	hns_cbusy_mtbit_exclude_rni	Exclude RNI sources in multi-source mode	RW	1'b0
[62:49]	Reserved	Reserved	RO	-
[48]	hns_cbusy_rd_wr_types_en	When set, CBusy for Reads and Writes are handled independently. The thresholds specified in this register are used for Read request types in POCQ	RW	1'b0
[47:24]	Reserved	Reserved	RO	-
[23:16]	hns_cbusy_high_limit	POCQ limit for CBusy High	RW	Configuration dependent
[15:8]	hns_cbusy_med_limit	POCQ limit for CBusy Med	RW	Configuration dependent
[7:0]	hns_cbusy_low_limit	POCQ limit for CBusy Low	RW	Configuration dependent

4.3.10.10 cmn_hns_txrsp_arb_weight_ctl

TXRSP arbitration weight controls.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA20`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-291: cmn_hns_txrsp_arb_weight_ctl

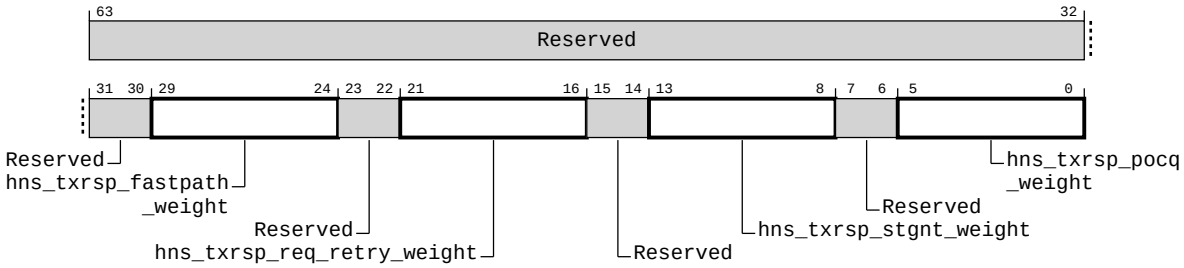


Table 4-307: cmn_hns_txrsp_arb_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	hns_txrsp_fastpath_weight	Fastpath response weights for TXRSP channel	RW	6'b111111
[23:22]	Reserved	Reserved	RO	-
[21:16]	hns_txrsp_req_retry_weight	Request retry response weights for TXRSP channel	RW	6'b000001
[15:14]	Reserved	Reserved	RO	-
[13:8]	hns_txrsp_stgnt_weight	Static Credit Grant response weights for TXRSP channel	RW	6'b000001
[7:6]	Reserved	Reserved	RO	-
[5:0]	hns_txrsp_pocq_weight	POCQ response weights for TXRSP channel	RW	6'b000001

4.3.10.11 cmn_hns_cbusy_mode_ctl

Control register for additional CBusy controls

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA28`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-292: cmn_hns_cbusy_mode_ctl

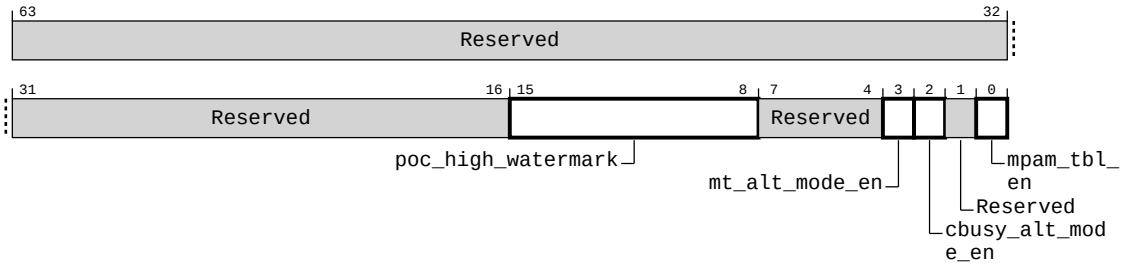


Table 4-308: cmn_hns_cbusy_mode_ctl attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	poc_high_watermark	Number of POCQ entries when it is considered high occupancy	RW	Configuration dependent
[7:4]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[3]	mt_alt_mode_en	Enable CBusy[2] alternate reporting mode: **1'b0** POCQ has requests from more than one source **1'b1** POCQ Occupancy is higher that the poc_high_watermark	RW	1'b0
[2]	cbusy_alt_mode_en	Enalbes an alternate mode of SN CBusy[1:0] capture for mpam_tbl_en=1 mode: **1'b0** For each MPAM partID, CBusy[1:0] = SN_CBusy[1:0] **1'b1** For each MPAM partID, CBusy[1] = SN's CBusy[2], CBusy[0] = (SN_CBusy[1] & SN_CBusy[0])	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	mpam_tbl_en	Enables cbusy reporting based on MPAM part ID	RW	1'b0

4.3.10.12 cmn_hns_lbt_cfg_ctl

Functions as the configuration control register for HN-F. Only applicable to LBT transactions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA30`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-293: cmn_hns_lbt_cfg_ctl

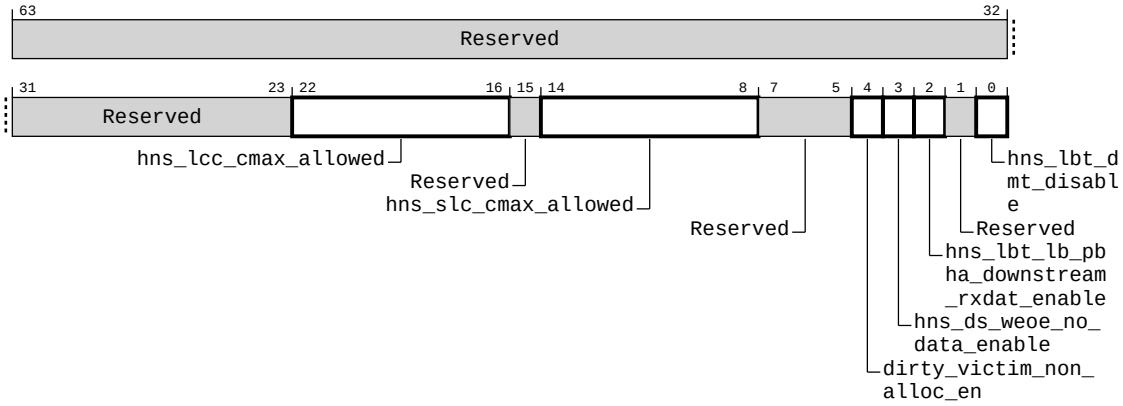


Table 4-309: cmn_hns_lbt_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:23]	Reserved	Reserved	RO	-
[22:16]	hns_lcc_cmax_allowed	Maximum cache capicity usage in fixed-point fraction of the cache capacity by LBT lines	RW	7'b1111111
[15]	Reserved	Reserved	RO	-
[14:8]	hns_slc_cmax_allowed	Maximum cache capicity usage in fixed-point fraction of the cache capacity by HBT lines	RW	7'b1111111
[7:5]	Reserved	Reserved	RO	-
[4]	dirty_victim_non_alloc_en	When HNS issues dirty CopyBack writes for LCC victim or SFBI, set non-allocating type	RW	1'b0
[3]	hns_ds_weoe_no_data_enable	When HNS issues WriteEvictOrEvict downstream, force no data transfer if config bit is set	RW	1'b0
[2]	hns_lbt_lb_pbha_downstream_rxdat_enable	Takes LB/PBHA values from downstream RXDAT for LBT lines	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	hns_lbt_dmt_disable	Disables DMT when set	RW	1'b0

4.3.10.13 cmn_hns_lbt_aux_ctl

Functions as the auxiliary control register for HN-F. Only applicable to LBT transactions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA38`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-294: cmn_hns_lbt_aux_ctl

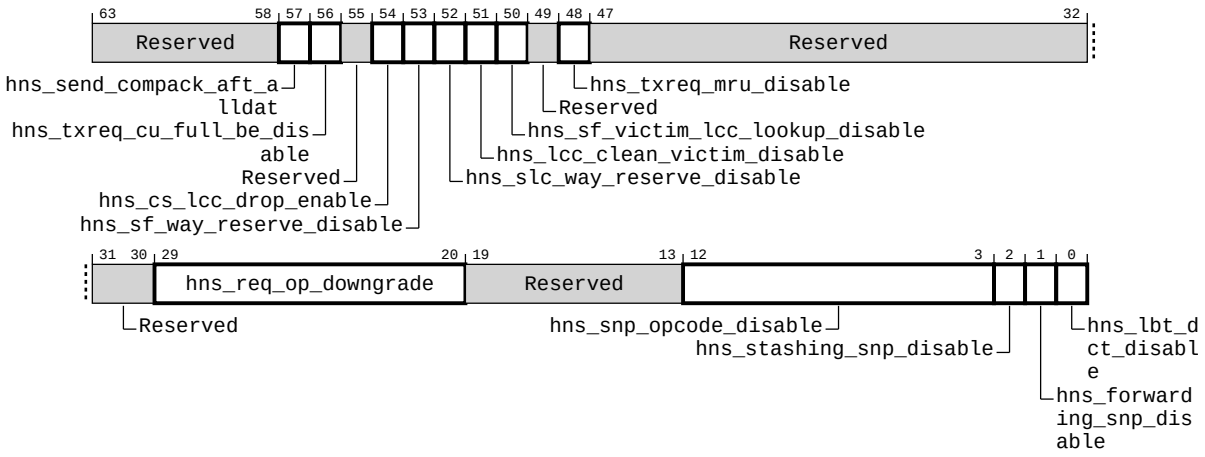


Table 4-310: cmn_hns_lbt_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57]	hns_send_compack_aft_alldat	Enables sending CompAck after all data beats are received	RW	1'b0
[56]	hns_txreq_cu_full_be_disable	Disables LCC always getting ownership by CleanUnique for streaming writes with full BE.	RW	1'b0
[55]	Reserved	Reserved	RO	-
[54]	hns_cs_lcc_drop_enable	Enables LCC to drop clean copy after writing dirty data for CleanShared.	RW	1'b1
[53]	hns_sf_way_reserve_disable	Disables SF reserved way for HBT lines in HNS mode and allow LBT lines to take all ways in SF.	RW	1'b0
[52]	hns_slc_way_reserve_disable	Disables SLC reserved way for HBT lines in HNS mode and allow LBT lines to take all ways in system cache.	RW	1'b0
[51]	hns_lcc_clean_victim_disable	Disables LCC sending clean eviction to Home Node.	RW	1'b0
[50]	hns_sf_victim_lcc_lookup_disable	Disables LCC lookup for a SF victim.	RW	1'b0
[49]	Reserved	Reserved	RO	-
[48]	hns_txreq_mru_disable	Disables sending MRU opcode downstream. Use RDUNIQ instead	RW	1'b0
[47:30]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[29:20]	hns_req_op_downgrade	Downgrades req opcode when set **[20]** Change READSHARED to READUNIQ **[21]** Change READNOTSHAREDDIRTY to READUNIQ **[22]** Change READPREFERUNIQ to READUNIQ **[23]** Change READONCEMKINV to READONCE **[24]** Change READONCECLNINV to READONCE **[25]** Change MAKEUNIQ to CLNUNIQ **[26]** Change MAKEREADUNIQ to READUNIQ **[27]** Change WRITEEVICTOREVICT to EVICT **[28]** Change MAKEINVALID to CLEANINVALID **[29]** Change WRITEUNIQUEFULL to non-allocating WRITEUNIQUEPTL	RW	10'h000
[19:13]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[12:3]	hns_snp_opcode_disable	Disables support for RXSNP different snoop opcodes and changes to similar snoop opcode when set **[3]** Change SNPSTASHSHARED to SNPQUERY **[4]** Change SNPSTASHUNIQUE to SNPQUERY **[5]** Change SNPMAKEINVALIDSTASH to SNPMAKEINVALID **[6]** Change SNPCLEANSHARED to SNPCLEANINVALID **[7]** Change SNPPREFERUNIQUE(FWD) to SNPUNIQUE **[8]** Change SNPNOTSHAREDDIRTY(FWD) to SNPUNIQUE **[9]** Change SNPCLEAN(FWD) to SNPUNIQUE **[10]** Change SNPUNIQUESTASH to SNPUNIQUE **[11]** Change SNPONCE(FWD) to SNPUNIQUE **[12]** Change SNPSHARED(FWD) to SNPUNIQUE	RW	10'h000
[2]	hns_stashing_snp_disable	Disables Stashing type of snoops when set for RXSNP	RW	1'b1
[1]	hns_forwarding_snp_disable	Disables Forwarding type of snoops when set for RXSNP	RW	1'b1
[0]	hns_lbt_dct_disable	Disables DCT when set	RW	1'b0

4.3.10.14 cmn_hns_ppu_pwpr

Functions as the power policy register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h1C00`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-295: cmn_hns_ppu_pwpr

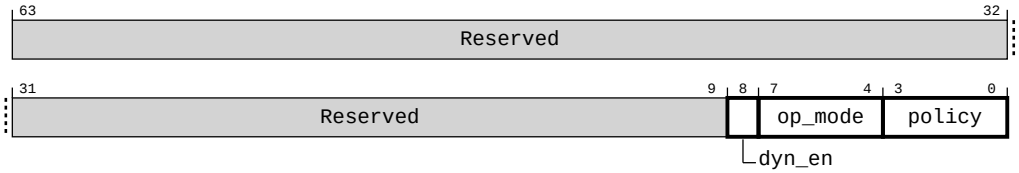


Table 4-311: cmn_hns_ppu_pwpr attributes

Bits	Name	Description	Type	Reset
[32:9]	Reserved	Reserved	RO	-
[8]	dyn_en	Dynamic transition enable	RW	1'b0
[7:4]	op_mode	HN-F operational power mode **4'b0011** FAM **4'b0010** HAM **4'b0001:** SFONLY **4'b0000** NOSFSLC	RW	4'b0
[3:0]	policy	HN-F power mode policy **4'b1000** ON **4'b0111** FUNC_RET **4'b0010:** MEM_RET **4'b0000** OFF	RW	4'b0

4.3.10.15 cmn_hns_ppu_pwsr

Provides power status information for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h1C08`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-296: cmn_hns_ppu_pwsr

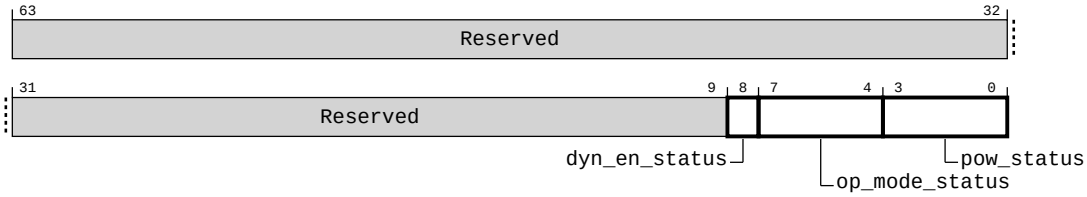


Table 4-312: cmn_hns_ppu_pwsr attributes

Bits	Name	Description	Type	Reset
[32:9]	Reserved	Reserved	RO	-
[8]	dyn_en_status	Dynamic transition status	RO	1'b0

Bits	Name	Description	Type	Reset
[7:4]	op_mode_status	HN-F operational mode status **4'b0011** FAM **4'b0010** HAM **4'b0001:** SFONLY **4'b0000** NOSFSLC	RO	4'b0
[3:0]	pow_status	HN-F power mode status **4'b1000** ON **4'b0111** FUNC_RET **4'b0010:** MEM_RET **4'b0000** OFF	RO	4'b0

4.3.10.16 cmn_hns_ppu_misr

Functions as the power miscellaneous input current status register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h1C14`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-297: cmn_hns_ppu_misr

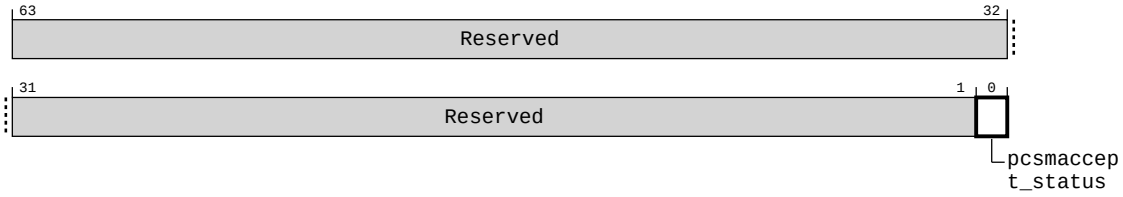


Table 4-313: cmn_hns_ppu_misr attributes

Bits	Name	Description	Type	Reset
[32:1]	Reserved	Reserved	RO	-
[0]	pcsmaccept_status	HN-F RAM PCSMACCEPT status	RO	1'b0

4.3.10.17 cmn_hns_ppu_idr0

Provides identification information for the HN-F PPU.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h2BB0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-298: cmn_hns_ppu_idr0

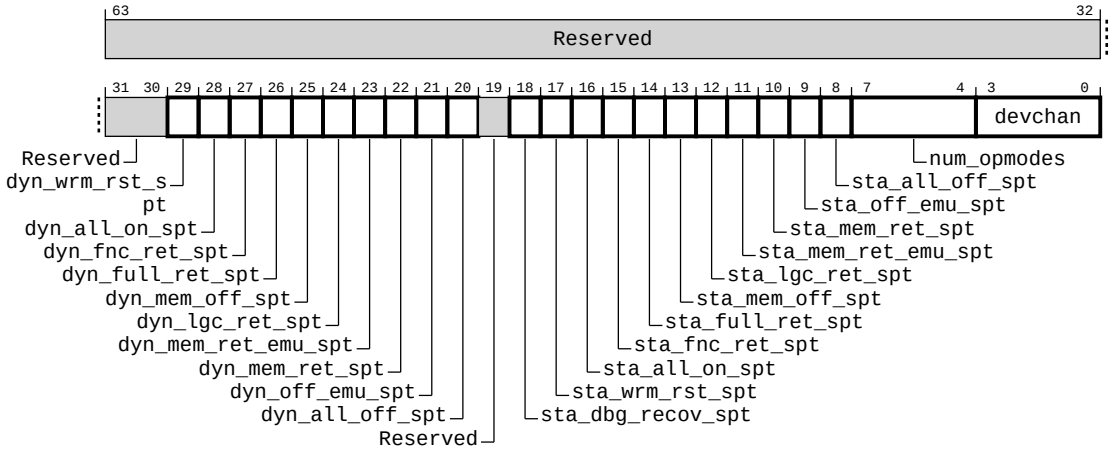


Table 4-314: cmn_hns_ppu_idr0 attributes

Bits	Name	Description	Type	Reset
[32:30]	Reserved	Reserved	RO	-
[29]	dyn_wrm_rst_spt	Dynamic warm_rst support	RO	1'b0
[28]	dyn_all_on_spt	Dynamic on support	RO	1'b0
[27]	dyn_fnc_ret_spt	Dynamic func_ret support	RO	1'b1
[26]	dyn_full_ret_spt	Dynamic full_ret support	RO	1'b0
[25]	dyn_mem_off_spt	Dynamic mem_off support	RO	1'b0
[24]	dyn_lgc_ret_spt	Dynamic logic_ret support	RO	1'b0
[23]	dyn_mem_ret_emu_spt	Dynamic mem_ret_emu support	RO	1'b0
[22]	dyn_mem_ret_spt	Dynamic mem_ret support	RO	1'b0
[21]	dyn_off_emu_spt	Dynamic off_emu support	RO	1'b0
[20]	dyn_all_off_spt	Dynamic off support	RO	1'b0
[19]	Reserved	Reserved	RO	-
[18]	sta_dbg_recov_spt	Static dbg_recov support	RO	1'b0
[17]	sta_wrm_rst_spt	Static warm_rst support	RO	1'b0
[16]	sta_all_on_spt	Static on support	RO	1'b1
[15]	sta_fnc_ret_spt	Static func_ret support	RO	1'b1
[14]	sta_full_ret_spt	Static full_ret support	RO	1'b0
[13]	sta_mem_off_spt	Static mem_off support	RO	1'b1
[12]	sta_lgc_ret_spt	Static logic_ret support	RO	1'b0
[11]	sta_mem_ret_emu_spt	Static mem_ret_emu support	RO	1'b0
[10]	sta_mem_ret_spt	Static mem_ret support	RO	1'b1
[9]	sta_off_emu_spt	Static off_emu support	RO	1'b0
[8]	sta_all_off_spt	Static off support	RO	1'b1
[7:4]	num_opmodes	Number of operational modes	RO	4'b0100
[3:0]	devchan	Number of device interface channels	RO	1'b0

4.3.10.18 cmn_hns_ppu_idr1

Provides identification information for the HN-F PPU.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h2BB4`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-299: cmn_hns_ppu_idr1

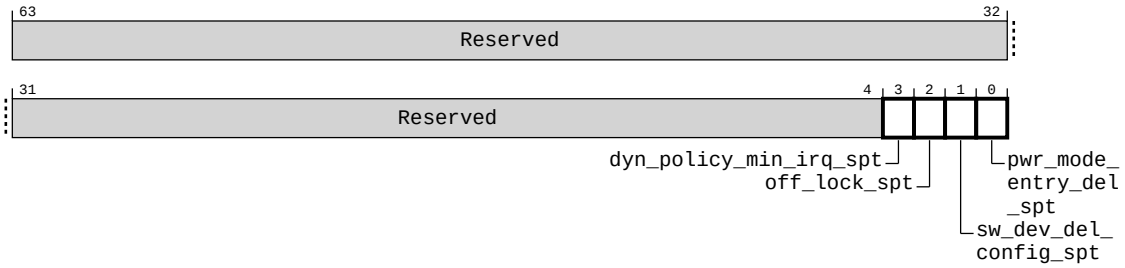


Table 4-315: cmn_hns_ppu_idr1 attributes

Bits	Name	Description	Type	Reset
[32:4]	Reserved	Reserved	RO	-
[3]	dyn_policy_min_irq_spt	Dynamic minimum policy interrupt support	RO	1'b0
[2]	off_lock_spt	Off and mem_ret lock support	RO	1'b0
[1]	sw_dev_del_config_spt	Software device delay control configuration support	RO	1'b0
[0]	pwr_mode_entry_del_spt	Power mode entry delay support	RO	1'b0

4.3.10.19 cmn_hns_ppu_iidr

Functions as the power implementation identification register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h2BC8`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-300: cmn_hns_ppu_iidr

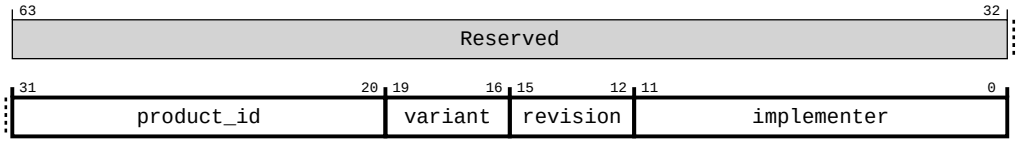


Table 4-316: cmn_hns_ppu_iidr attributes

Bits	Name	Description	Type	Reset
[32]	Reserved	Reserved	RO	-
[31:20]	product_id	Implementation identifier	RO	12'h434
[19:16]	variant	Implementation variant	RO	4'h0
[15:12]	revision	Implementation revision	RO	4'h0
[11:0]	implementer	Arm implementation	RO	12'h43B

4.3.10.20 cmn_hns_ppu_aidr

Functions as the power architecture identification register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

`16'h2BCC`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-301: cmn_hns_ppu_aidr

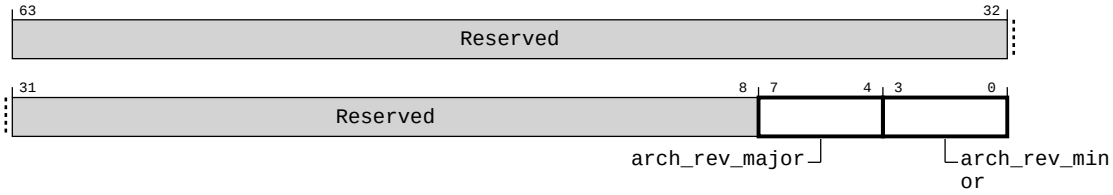


Table 4-317: cmn_hns_ppu_aidr attributes

Bits	Name	Description	Type	Reset
[32:8]	Reserved	Reserved	RO	-
[7:4]	arch_rev_major	PPU architecture major revision	RO	4'h1
[3:0]	arch_rev_minor	PPU architecture minor revision	RO	4'h1

4.3.10.21 cmn_hns_ppu_dyn_ret_threshold

Configures the dynamic retention threshold for SLC and SF RAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1D00`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-302: cmn_hns_ppu_dyn_ret_threshold

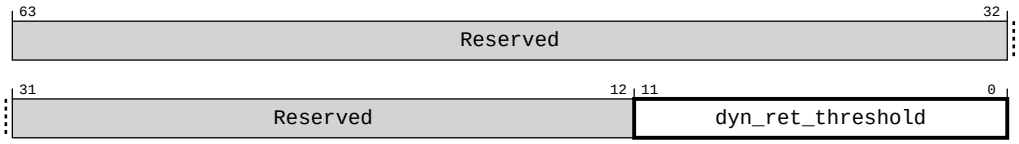


Table 4-318: cmn_hns_ppu_dyn_ret_threshold attributes

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	-
[11:0]	dyn_ret_threshold	HN-F RAM idle cycle count threshold	RW	32'b0

4.3.10.22 cmn_hns_qos_band

Provides QoS classifications based on the QoS value ranges.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA80

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-303: cmn_hns_qos_band

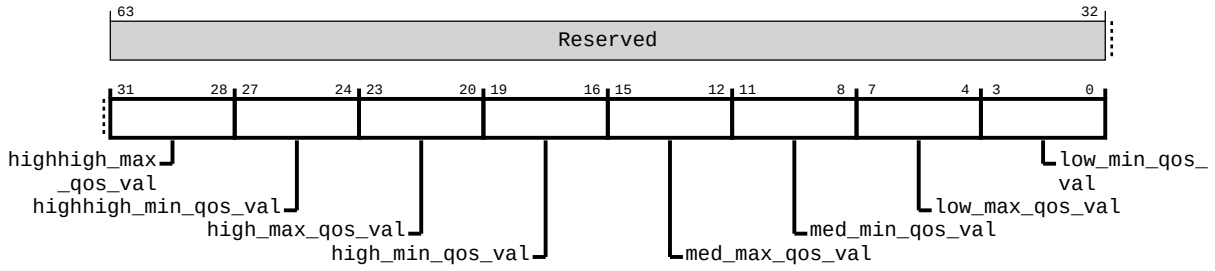


Table 4-319: cmn_hns_qos_band attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	highhigh_max_qos_val	Maximum value for HighHigh QoS class	RO	4'hF
[27:24]	highhigh_min_qos_val	Minimum value for HighHigh QoS class	RO	4'hF
[23:20]	high_max_qos_val	Maximum value for High QoS class	RO	4'hE
[19:16]	high_min_qos_val	Minimum value for High QoS class	RO	4'hC

Bits	Name	Description	Type	Reset
[15:12]	med_max_qos_val	Maximum value for Medium QoS class	RO	4'hB
[11:8]	med_min_qos_val	Minimum value for Medium QoS class	RO	4'h8
[7:4]	low_max_qos_val	Maximum value for Low QoS class	RO	4'h7
[3:0]	low_min_qos_val	Minimum value for Low QoS class	RO	4'h0

4.3.10.23 cmn_hns_errfr

Functions as the error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3000`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-304: cmn_hns_errfr

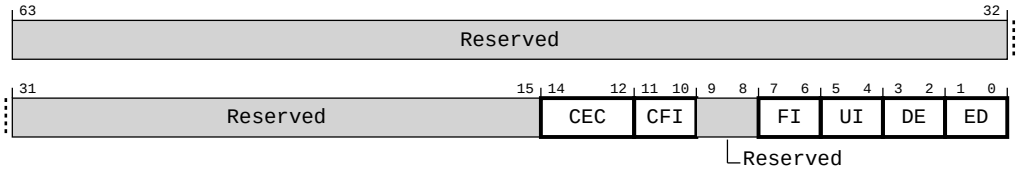


Table 4-320: cmn_hns_errfr attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:12]	CEC	Standard corrected error count mechanism **3'b000** Does not implement standardized error counter model **3'b010** Implements 8-bit error counter in cmn_hns_errmisc[39:32] **3'b100** Implements 16-bit error counter in cmn_hns_errmisc[47:32]	RO	3'b100
[11:10]	CFI	Corrected error interrupt	RO	2'b10
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors for data poison	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.10.24 cmn_hns_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3008`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-305: cmn_hns_errctlr

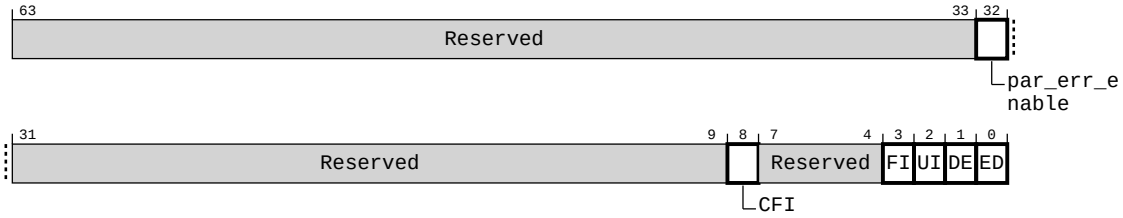


Table 4-321: cmn_hns_errctlr attributes

Bits	Name	Description	Type	Reset
[63:33]	Reserved	Reserved	RO	-
[32]	par_err_enable	Enables external logging parity errors when set to 1'b1	RW	1'b0
[31:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in cmn_hns_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in cmn_hns_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in cmn_hns_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in cmn_hns_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in cmn_hns_errfr.ED	RW	1'b0

4.3.10.25 cmn_hns_errstatus

Functions as the error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3010`

Type

W1C

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-306: cmn_hns_errstatus

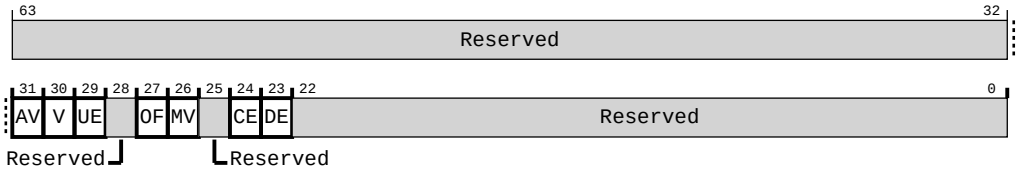


Table 4-322: cmn_hns_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; cmn_hns_erraddr contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0

Bits	Name	Description	Type	Reset
[26]	MV	cmn_hns_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.10.26 cmn_hns_erraddr

Contains the error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3018`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-307: cmn_hns_erraddr

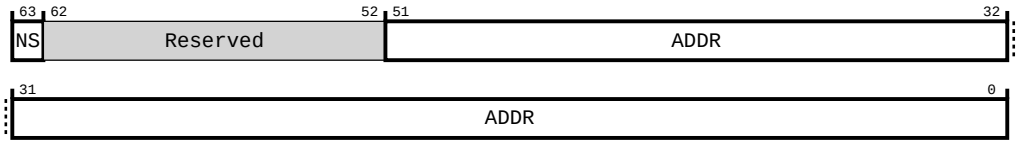


Table 4-323: cmn_hns_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: cmn_hns_erraddr.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.10.27 cmn_hns_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3020`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-308: cmn_hns_errmisc

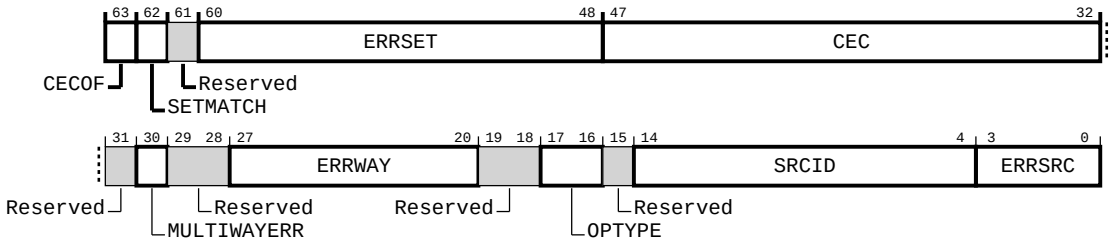


Table 4-324: cmn_hns_errmisc attributes

Bits	Name	Description	Type	Reset
[63]	CECOF	Corrected error counter overflow	RW	1'b0
[62]	SETMATCH	Set address match	RW	1'b0
[61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	SLC/SF set address for ECC error	RW	13'b0
[47:32]	CEC	Corrected ECC error count	RW	16'b0
[31]	Reserved	Reserved	RO	-
[30]	MULTIWAYERR	Indicate multiple ways have ECC error	RW	1'b0
[29:28]	Reserved	Reserved	RO	-
[27:20]	ERRWAY	SLC/SF way ID for ECC error	RW	8'b0
[19:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error op type **2'b00** Writes, CleanShared, Atomics and stash requests with invalid targets **2'b01** WriteBack, Evict, and Stash requests with valid target **2'b10** CMO **2'b11** Other op types	RW	2'b00
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	Error source **4'b0001** Data single-bit ECC **4'b0010** Data double-bit ECC **4'b0011** Single-bit ECC overflow **4'b0100** Tag single-bit ECC **4'b0101** Tag double-bit ECC **4'b0111** SF tag single-bit ECC **4'b1000** SF tag double-bit ECC **4'b1010** Data parity error **4'b1011** Data parity and poison **4'b1100** NDE	RW	4'b0000

4.3.10.28 cmn_hns_err_inj

Enables error injection and setup. When enabled for a given source ID and logic processor ID, HN-F returns a subordinate error and reports an error interrupt. This error interrupt emulates a SLC double-bit data ECC error. This feature enables software to test the error handler. The subordinate error is reported for cacheable read access for which SLC hit is the data source. No subordinate error or error interrupt is reported for cacheable read access in which SLC miss is the data source.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3030`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-309: cmn_hns_err_inj

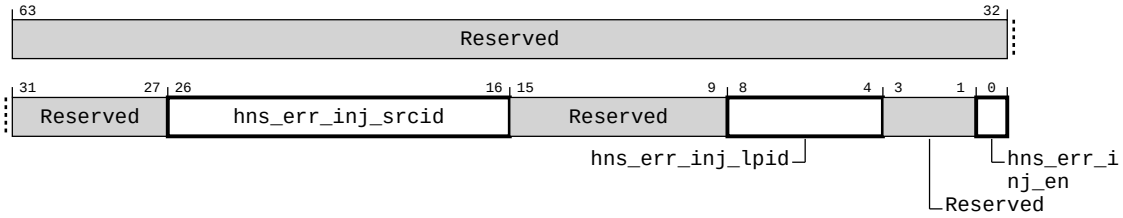


Table 4-325: cmn_hns_err_inj attributes

Bits	Name	Description	Type	Reset
[63:27]	Reserved	Reserved	RO	-
[26:16]	hns_err_inj_srcid	RN source ID for read access which results in a SLC miss; does not report subordinate error or error to match error injection	RW	11'h0
[15:9]	Reserved	Reserved	RO	-
[8:4]	hns_err_inj_lpid	LPID used to match for error injection	RW	5'h0
[3:1]	Reserved	Reserved	RO	-
[0]	hns_err_inj_en	Enables error injection and report	RW	1'b0

4.3.10.29 cmn_hns_byte_par_err_inj

Functions as the byte parity error injection register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3038`

Type

WO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-310: cmn_hns_byte_par_err_inj

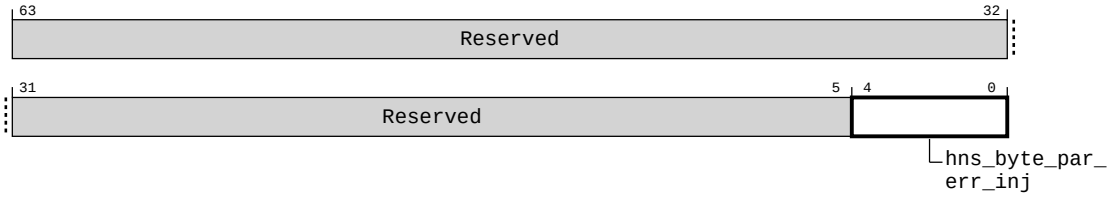


Table 4-326: cmn_hns_byte_par_err_inj attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:0]	hns_byte_par_err_inj	Specifies a byte lane; once this register is written, a byte parity error is injected in the specified byte lane on the next SLC hit; the error will be injected in all data flits on specified byte (0 to 31)	WO	5'h0

4.3.10.30 cmn_hns_errfr_NS

Functions as the Non-secure error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3100`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-311: cmn_hns_errfr_NS

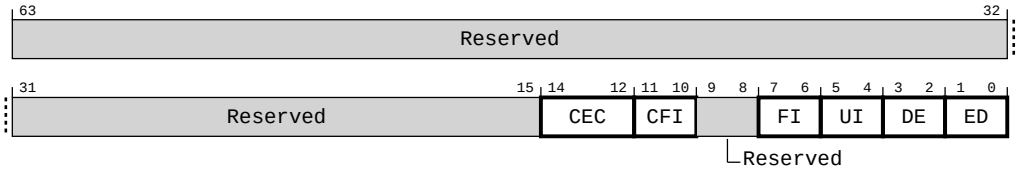


Table 4-327: cmn_hns_errfr_NS attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism **3'b000** Does not implement standardized error counter model **3'b010** Implements 8-bit error counter in cmn_hns_errmisc[39:32] **3'b100** Implements 16-bit error counter in cmn_hns_errmisc[47:32]	RO	3'b100
[11:10]	CFI	Corrected error interrupt	RO	2'b10
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors for data poison	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.10.31 cmn_hns_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3108`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-312: cmn_hns_errctlr_NS

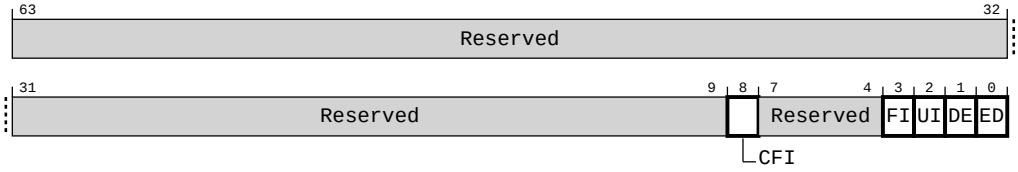


Table 4-328: cmn_hns_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in cmn_hns_errfr_NS.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in cmn_hns_errfr_NS.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in cmn_hns_errfr_NS.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in cmn_hns_errfr_NS.DE	RW	1'b0
[0]	ED	Enables error detection as specified in cmn_hns_errfr_NS.ED	RW	1'b0

4.3.10.32 cmn_hns_errstatus_NS

Functions as the Non-secure error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3110`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-313: cmn_hns_errstatus_NS

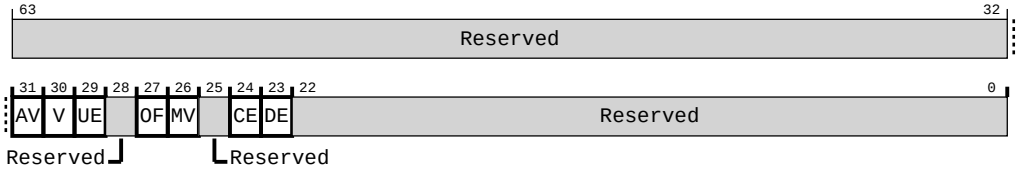


Table 4-329: cmn_hns_errstatus_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; cmn_hns_erraddr_NS contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	cmn_hns_errmisc_NS valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0

Bits	Name	Description	Type	Reset
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.10.33 cmn_hns_erraddr_NS

Contains the Non-secure error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3118`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-314: cmn_hns_erraddr_NS

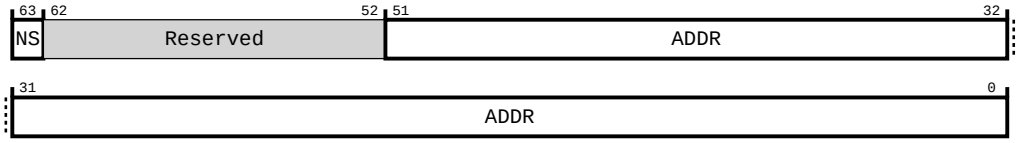


Table 4-330: cmn_hns_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: cmn_hns_erraddr_NS.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.10.34 cmn_hns_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3120`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-315: cmn_hns_errmisc_NS

Table 4-331: cmn_hns_errmisc_NS attributes

Bits	Name	Description	Type	Reset
[63]	CECOF	Corrected error counter overflow	RW	1'b0
[62]	SETMATCH	Set address match	RW	1'b0
[61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	SLC/SF set address for ECC error	RW	13'b0
[47:32]	CEC	Corrected ECC error count	RW	16'b0
[31]	Reserved	Reserved	RO	-
[30]	MULTIWAYERR	Indicate multiple ways have ECC error	RW	1'b0
[29:28]	Reserved	Reserved	RO	-
[27:20]	ERRWAY	SLC/SF way ID for ECC error	RW	8'b0
[19:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error op type **2'b00** Writes, CleanShared, Atomics and stash requests with invalid targets **2'b01** WriteBack, Evict, and Stash requests with valid target **2'b10** CMO **2'b11** Other op types	RW	2'b00
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0
[3:0]	ERRSRC	Error source **4'b0001** Data single-bit ECC **4'b0010** Data double-bit ECC **4'b0011** Single-bit ECC overflow **4'b0100** Tag single-bit ECC **4'b0101** Tag double-bit ECC **4'b0111** SF tag single-bit ECC **4'b1000** SF tag double-bit ECC **4'b1010** Data parity error **4'b1011** Data parity and poison **4'b1100** NDE	RW	4'b0000

4.3.10.35 cmn_hns_slc_lock_ways

Controls SLC way lock settings.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses. The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-316: cmn_hns_slc_lock_ways

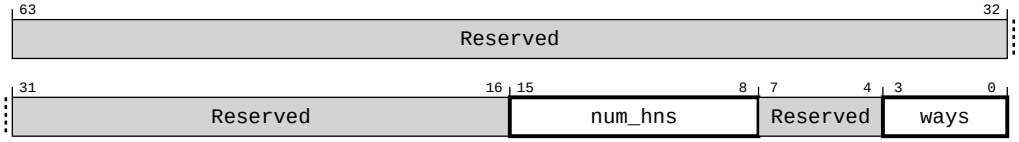


Table 4-332: cmn_hns_slc_lock_ways attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	num_hns	Number of HN-Fs in NUMA (non-uniform memory access) region	RW	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3:0]	ways	Number of SLC ways locked (1, 2, 4, 8, 12)	RW	4'b0

4.3.10.36 cmn_hns_slc_lock_base0

Functions as the base register for lock region 0 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC08`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses. The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-317: cmn_hns_slc_lock_base0

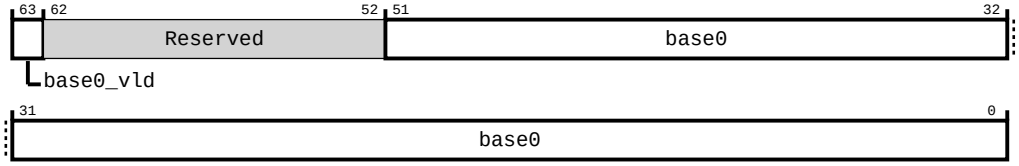


Table 4-333: cmn_hns_slc_lock_base0 attributes

Bits	Name	Description	Type	Reset
[63]	base0_vld	Lock region 0 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base0	Lock region 0 base address	RW	52'b0

4.3.10.37 cmn_hns_slc_lock_base1

Functions as the base register for lock region 1 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC10`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses. The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-318: cmn_hns_slc_lock_base1

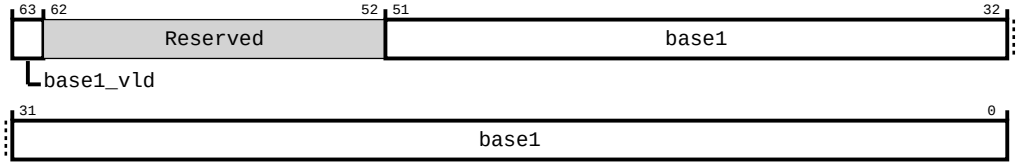


Table 4-334: cmn_hns_slc_lock_base1 attributes

Bits	Name	Description	Type	Reset
[63]	base1_vld	Lock region 1 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base1	Lock region 1 base address	RW	52'b0

4.3.10.38 cmn_hns_slc_lock_base2

Functions as the base register for lock region 2 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC18`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses. The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-319: cmn_hns_slc_lock_base2

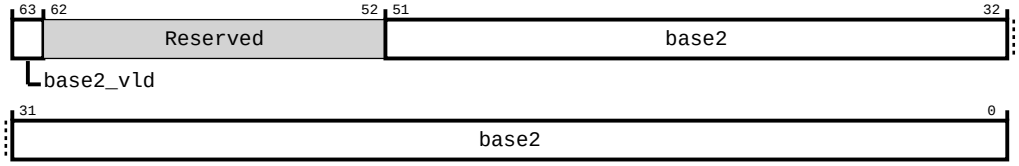


Table 4-335: cmn_hns_slc_lock_base2 attributes

Bits	Name	Description	Type	Reset
[63]	base2_vld	Lock region 2 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base2	Lock region 2 base address	RW	52'b0

4.3.10.39 cmn_hns_slc_lock_base3

Functions as the base register for lock region 3 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC20`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses. The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-320: cmn_hns_slc_lock_base3

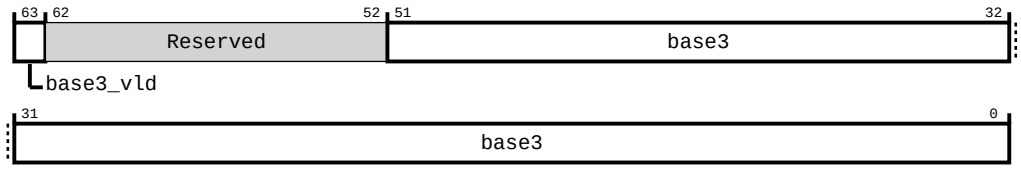


Table 4-336: cmn_hns_slc_lock_base3 attributes

Bits	Name	Description	Type	Reset
[63]	base3_vld	Lock region 3 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base3	Lock region 3 base address	RW	52'b0

4.3.10.40 cmn_hns_rni_region_vec

Functions as the control register for RN-I source SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC28`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-321: cmn_hns_rni_region_vec

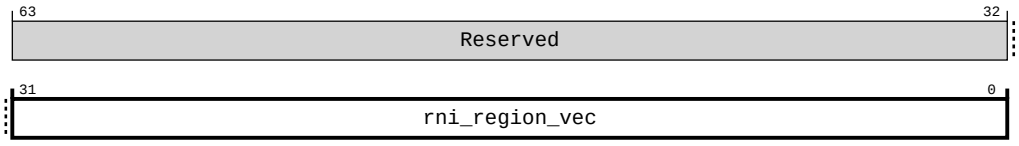


Table 4-337: cmn_hns_rni_region_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_region_vec	Bit vector mask; identifies which logical IDs of the RN-Is to allocate to the locked region NOTE: Must be set to 32'b0 if range-based region locking or OCM is enabled.	RW	32'b0

4.3.10.41 cmn_hns_rnd_region_vec

Functions as the control register for RN-D source SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC30`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-322: cmn_hns_rnd_region_vec

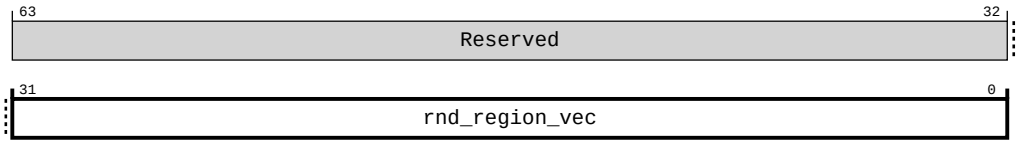


Table 4-338: cmn_hns_rnd_region_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_region_vec	Bit vector mask; identifies which logical IDs of the RN-Ds to allocate to the locked region NOTE: Must be set to 32'b0 if range-based region locking or OCM is enabled.	RW	32'b0

4.3.10.42 cmn_hns_rnf_region_vec

Functions as the control register for RN-F source SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC38`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-323: cmn_hns_rnf_region_vec

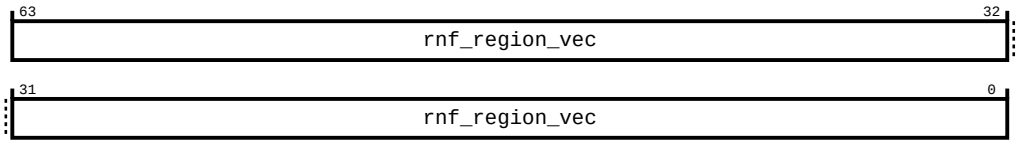


Table 4-339: cmn_hns_rnf_region_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_region_vec	Bit vector mask; identifies which logical IDs of the RN-Fs to allocate to the locked region NOTE: Must be 64'b0 if range-based region locking or OCM is enabled.	RW	64'b0

4.3.10.43 cmn_hns_rnf_region_vec1

Functions as the control register for RN-F source SLC way allocation for logical IDs 64 through 127.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC40`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-324: cmn_hns_rnf_region_vec1

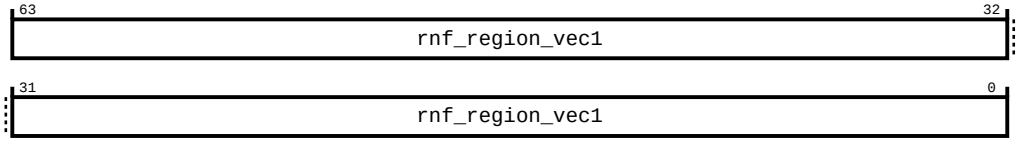


Table 4-340: cmn_hns_rnf_region_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_region_vec1	Bit vector mask; identifies which logical IDs of the RN-Fs to allocate to the locked region NOTE: Must be 64'b0 if range-based region locking or OCM is enabled.	RW	64'b0

4.3.10.44 cmn_hns_slcway_partition0_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC48`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-325: cmn_hns_slcway_partition0_rnf_vec

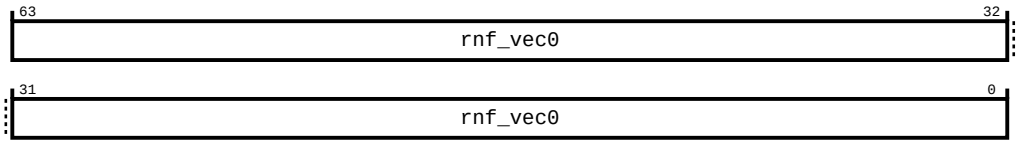


Table 4-341: cmn_hns_slcway_partition0_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec0	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.45 cmn_hns_slcway_partition1_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 1 (ways 4, 5, 6, and 7).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC50`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-326: cmn_hns_slcway_partition1_rnf_vec

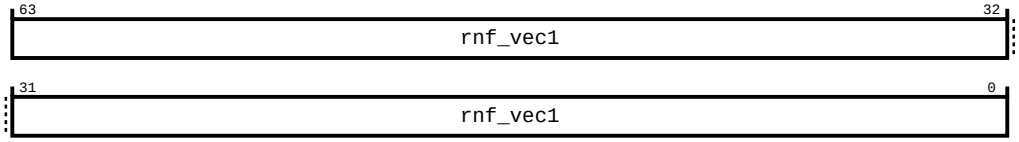


Table 4-342: cmn_hns_slcway_partition1_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec1	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.46 cmn_hns_slcway_partition2_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 2 (ways 8, 9, 10, and 11).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC58`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-327: cmn_hns_slcway_partition2_rnf_vec

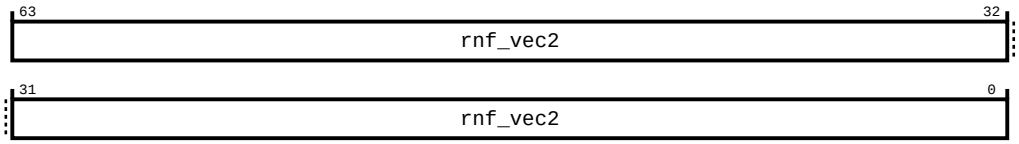


Table 4-343: cmn_hns_slcway_partition2_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec2	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.47 cmn_hns_slcway_partition3_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 3 (ways 12, 13, 14, and 15).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC60

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-328: cmn_hns_slcway_partition3_rnf_vec

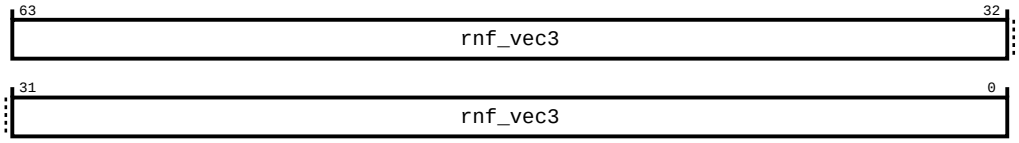


Table 4-344: cmn_hns_slcway_partition3_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec3	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.48 cmn_hns_slcway_partition0_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCB0

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-329: cmn_hns_slcway_partition0_rnf_vec1

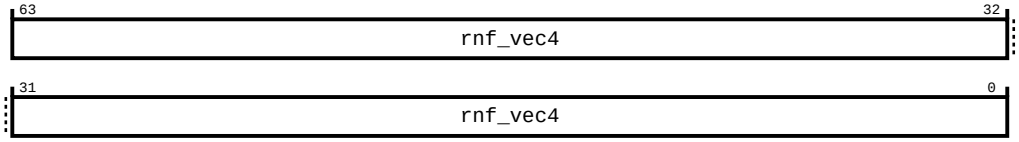


Table 4-345: cmn_hns_slcway_partition0_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec4	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.49 cmn_hns_slcway_partition1_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 1 (ways 4, 5, 6, and 7) for Logical RNF IDs 64 to 127.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCB8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-330: cmn_hns_slcway_partition1_rnf_vec1

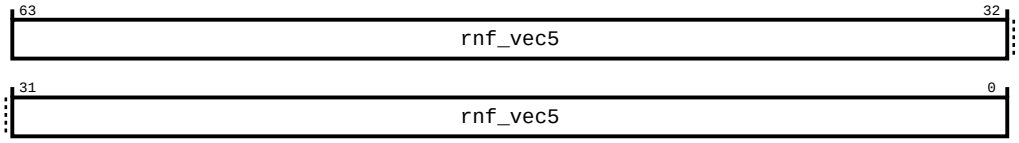


Table 4-346: cmn_hns_slcway_partition1_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec5	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.50 cmn_hns_slcway_partition2_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 2 (ways 8, 9, 10, and 11) for Logical RNF IDs 64 to 127.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCC0`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-331: cmn_hns_slcway_partition2_rnf_vec1

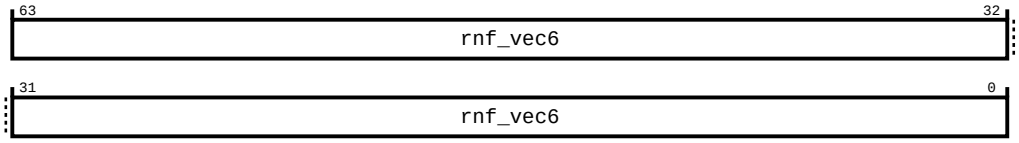


Table 4-347: cmn_hns_slcway_partition2_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec6	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.51 cmn_hns_slcway_partition3_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 3 (ways 12, 13, 14, and 15) for Logical RNF IDs 64 to 127.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCC8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-332: cmn_hns_slcway_partition3_rnf_vec1

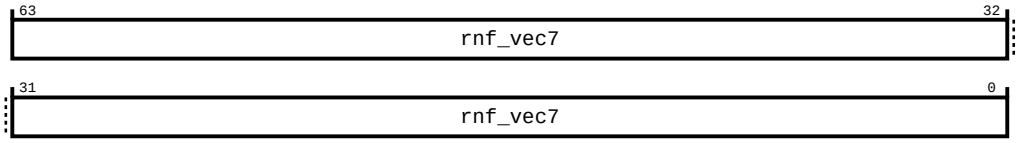


Table 4-348: cmn_hns_slcway_partition3_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec7	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

4.3.10.52 cmn_hns_slcway_partition0_rni_vec

Functions as the control register for RN-Is that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC68`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-333: cmn_hns_slcway_partition0_rni_vec

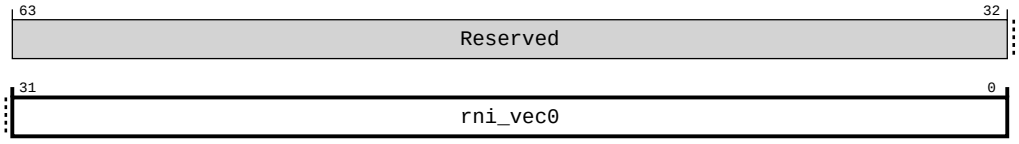


Table 4-349: cmn_hns_slcway_partition0_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec0	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.53 cmn_hns_slcway_partition1_rni_vec

Functions as the control register for RN-Is that can allocate to partition 1 (ways 4, 5, 6, and 7).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC70`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-334: cmn_hns_slcway_partition1_rni_vec

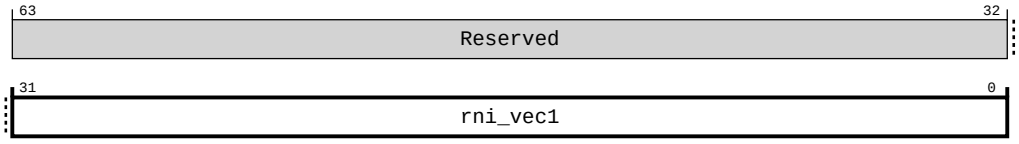


Table 4-350: cmn_hns_slcway_partition1_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec1	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.54 cmn_hns_slcway_partition2_rni_vec

Functions as the control register for RN-Is that can allocate to partition 2 (ways 8, 9, 10, and 11).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC78`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-335: cmn_hns_slcway_partition2_rni_vec

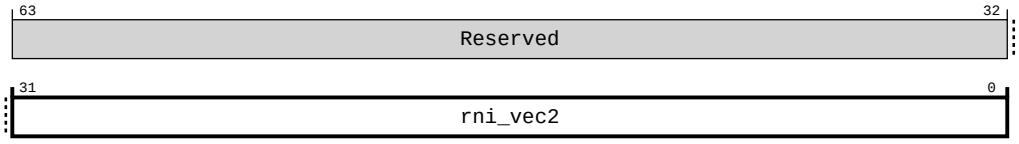


Table 4-351: cmn_hns_slcway_partition2_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec2	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.55 cmn_hns_slcway_partition3_rni_vec

Functions as the control register for RN-Is that can allocate to partition 3 (ways 12, 13, 14, and 15).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC80`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-336: cmn_hns_slcway_partition3_rni_vec

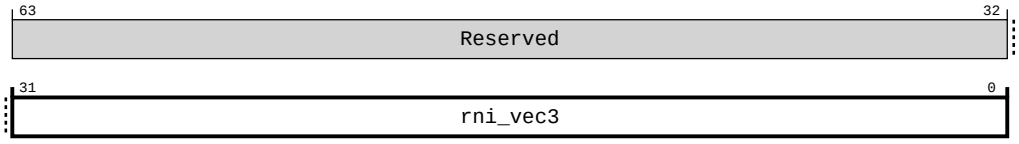


Table 4-352: cmn_hns_slcway_partition3_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec3	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.56 cmn_hns_slcway_partition0_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC88`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-337: cmn_hns_slcway_partition0_rnd_vec

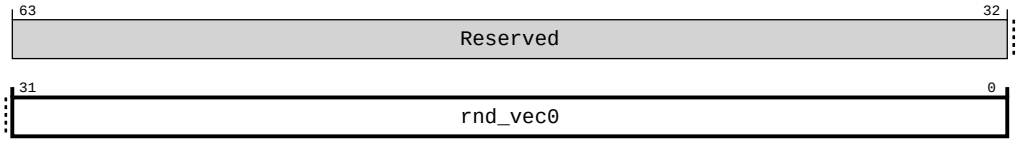


Table 4-353: cmn_hns_slcway_partition0_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec0	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.57 cmn_hns_slcway_partition1_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 1 (ways 4, 5, 6, and 7).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC90`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-338: cmn_hns_slcway_partition1_rnd_vec

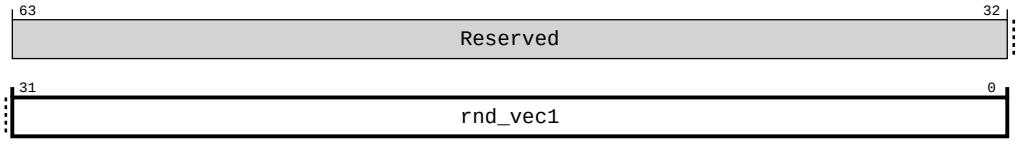


Table 4-354: cmn_hns_slcway_partition1_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec1	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.58 cmn_hns_slcway_partition2_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 2 (ways 8, 9, 10, and 11).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC98`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-339: cmn_hns_slcway_partition2_rnd_vec

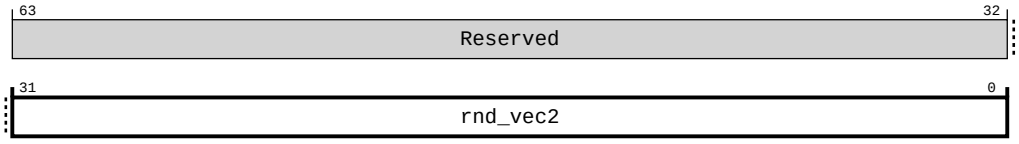


Table 4-355: cmn_hns_slcway_partition2_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec2	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.59 cmn_hns_slcway_partition3_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 3 (ways 12, 13, 14, and 15).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCA0`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-340: cmn_hns_slcway_partition3_rnd_vec

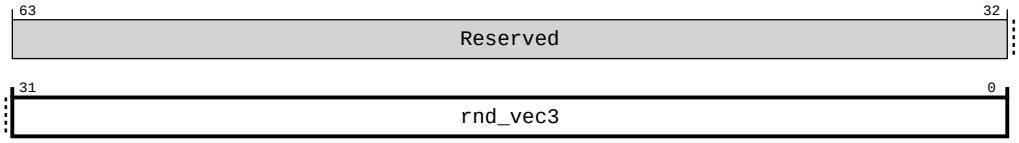


Table 4-356: cmn_hns_slcway_partition3_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec3	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

4.3.10.60 cmn_hns_rn_region_lock

Functions as the enable register for source-based SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCA8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-341: cmn_hns_rn_region_lock

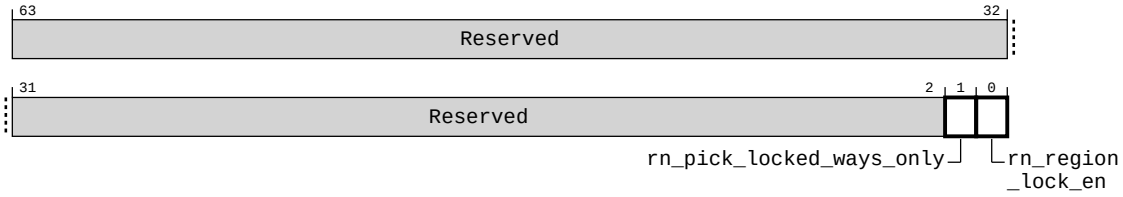


Table 4-357: cmn_hns_rn_region_lock attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	rn_pick_locked_ways_only	Specifies which ways the programmed RNs can allocate new cache lines to **1'b0** Programmed RN will choose all ways including locked **1'b1** Programmed RN will only allocate in locked ways	RW	1'b0
[0]	rn_region_lock_en	Enables SRC-based region locking **1'b0** SRC based way locking is disabled **1'b1** SRC based way locking is enabled	RW	1'b0

4.3.10.61 cmn_hns_sf_cxg_blocked_ways

Specifies the SF ways that are blocked for remote chip to use in CML mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCD0

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-342: cmn_hns_sf_cxg_blocked_ways

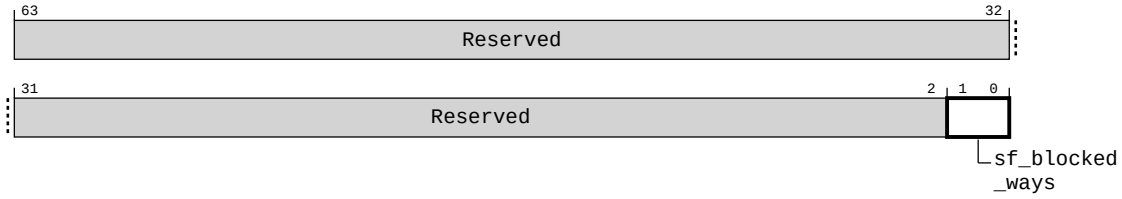


Table 4-358: cmn_hns_sf_cxg_blocked_ways attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	sf_blocked_ways	Number of SF ways blocked from remote chips to be able to use in CML mode. **2'b00** No ways are blocked; all SF ways could be used by local or remote RN-Fs **2'b01** SF_NUM_WAYS = 16: ways 3:0 for local RN-Fs only; ways 15:4 for local and remote RN-Fs SF_NUM_WAYS > 16: ways 7:0 for local RN-Fs only; ways 31:8 for locan and remote RN-Fs **2'b10** SF_NUM_WAYS = 16: ways 7:0 for local RN-Fs only; ways 15:8 for local and remote RN-Fs SF_NUM_WAYS > 16: ways 15:0 for local RN-Fs only; ways 31:16 for locan and remote RN-Fs **2'b11** SF_NUM_WAYS < 26: ways 11:0 for local RN-Fs only; ways 15:12 for local and remote RN-Fs SF_NUM_WAYS >= 26: ways 23:0 for local RN-Fs only; ways 31:24 for locan and remote RN-Fs	RW	2'b00

4.3.10.62 cmn_hns_cxg_ha_metadata_exclusion_list

Functions as the control register to identify CXG HA which does not support metadata

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCE0

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-343: cmn_hns_cxg_ha_metadata_exclusion_list

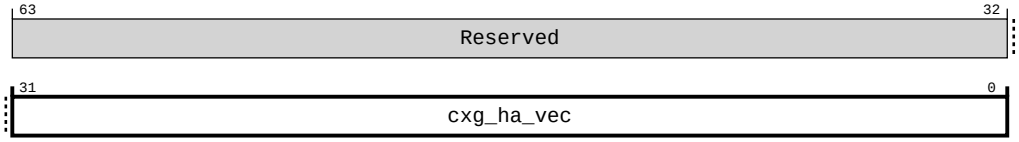


Table 4-359: cmn_hns_cxg_ha_metadata_exclusion_list attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	cxg_ha_vec	Bit vector mask; identifies which logical IDs of the CXG HA does not support metadata	RW	32'h00000000

4.3.10.63 cmn_hns_cxg_ha_smp_exclusion_list

Functions as the control register to identify CXG HA not connected to SMP CCIX link

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCD8

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-344: cmn_hns_cxg_ha_smp_exclusion_list

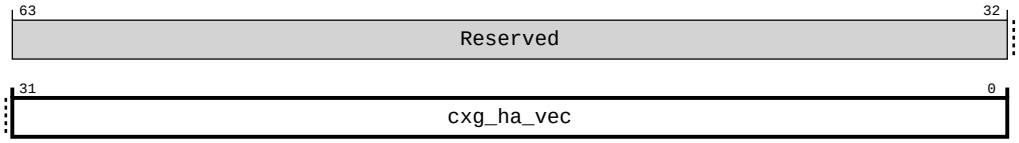


Table 4-360: cmn_hns_cxg_ha_smp_exclusion_list attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	cxg_ha_vec	Bit vector mask; identifies which logical IDs of the CXG HA does not connect to SMP CCIX link	RW	32'h00000000

4.3.10.64 hn_sam_hash_addr_mask_reg

Configures the address mask that is applied before hashing the address bits.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCF0

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-345: hn_sam_hash_addr_mask_reg

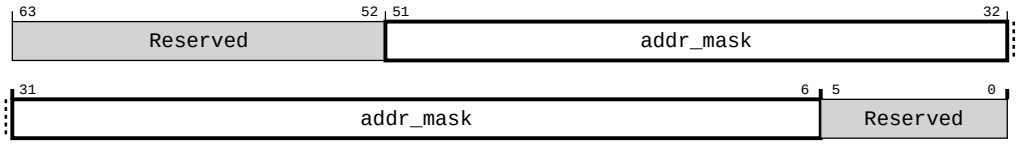


Table 4-361: hn_sam_hash_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address mask applied before hashing	RW	46'h3FFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

4.3.10.65 hn_sam_region_cmp_addr_mask_reg

Configures the address mask that is applied before memory region compare.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCF8`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-346: hn_sam_region_cmp_addr_mask_reg

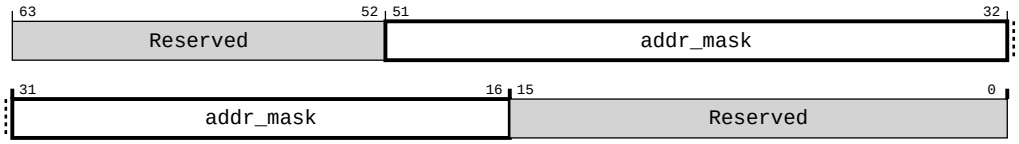


Table 4-362: hn_sam_region_cmp_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	addr_mask	Address mask applied before memory region compare	RW	36'hFFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

4.3.10.66 cmn_hns_sam_cfg1_def_hashed_region

Configures default hashed region in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD48`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-347: cmn_hns_sam_cfg1_def_hashed_region

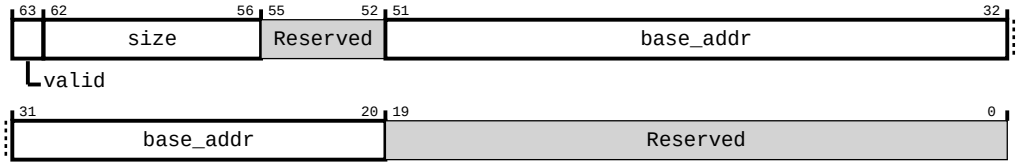


Table 4-363: cmn_hns_sam_cfg1_def_hashed_region attributes

Bits	Name	Description	Type	Reset
[63]	valid	Default hashed region valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'h1
[62:56]	size	Default hashed region 0 size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'h7F
[55:52]	Reserved	Reserved	RO	-
[51:20]	base_addr	Bits [51:20] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

4.3.10.67 cmn_hns_sam_cfg2_def_hashed_region

Configures default hashed region in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD50`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-348: cmn_hns_sam_cfg2_def_hashed_region

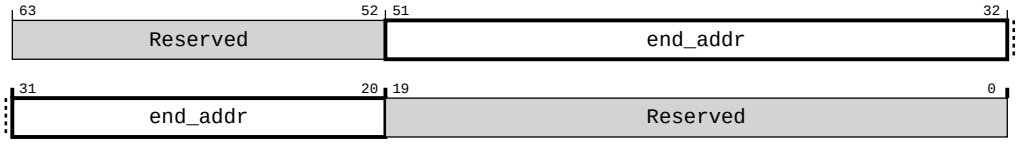


Table 4-364: cmn_hns_sam_cfg2_def_hashed_region attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	end_addr	Bits [51:20] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'hFFFFFFFF
[19:0]	Reserved	Reserved	RO	-

4.3.10.68 cmn_hns_sam_control

Configures HN-F SAM. All top_address_bit fields must be between bits 47 and 28 of the address. top_address_bit2 > top_address_bit1 > top_address_bit0. Must be configured to match corresponding por_rnsam_sys_cache_grp_sn_sam_cfgN register in the RN SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD00`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-349: cmn_hns_sam_control

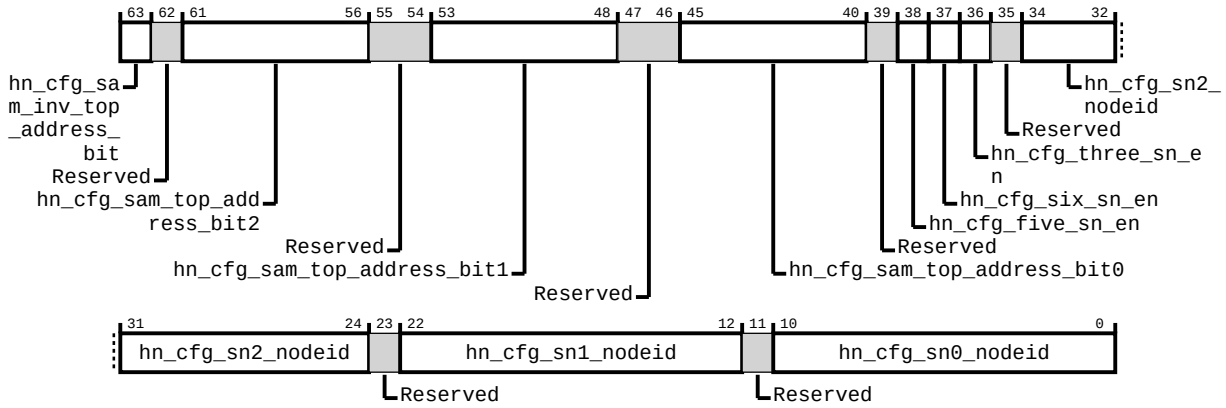


Table 4-365: cmn_hns_sam_control attributes

Bits	Name	Description	Type	Reset
[63]	hn_cfg_sam_inv_top_address_bit	Inverts the top address bit (hn_cfg_sam_top_address_bit1 if 3-SN, hn_cfg_sam_top_address_bit2 if 6-SN) NOTE: Can only be used when the address map does not have unique address bit combinations.	RW	1'h0
[62]	Reserved	Reserved	RO	-
[61:56]	hn_cfg_sam_top_address_bit2	Bit position of top_address_bit2; used for address hashing in 6-SN configuration	RW	6'h00
[55:54]	Reserved	Reserved	RO	-
[53:48]	hn_cfg_sam_top_address_bit1	Bit position of top_address_bit1; used for address hashing in 3-SN/6-SN configuration	RW	6'h00
[47:46]	Reserved	Reserved	RO	-
[45:40]	hn_cfg_sam_top_address_bit0	Bit position of top_address_bit0; used for address hashing in 3-SN/6-SN configuration	RW	6'h00
[39]	Reserved	Reserved	RO	-
[38]	hn_cfg_five_sn_en	Enables 5-SN configuration	RW	1'b0
[37]	hn_cfg_six_sn_en	Enables 6-SN configuration	RW	1'b0
[36]	hn_cfg_three_sn_en	Enables 3-SN configuration	RW	1'b0
[35]	Reserved	Reserved	RO	-
[34:24]	hn_cfg_sn2_nodeid	SN 2 node ID	RW	11'h0
[23]	Reserved	Reserved	RO	-
[22:12]	hn_cfg_sn1_nodeid	SN 1 node ID	RW	11'h0
[11]	Reserved	Reserved	RO	-
[10:0]	hn_cfg_sn0_nodeid	SN 0 node ID	RW	11'h0

4.3.10.69 cmn_hns_sam_control2

Configures HN-F SAM. Must be configured to match corresponding por_rnsam_sys_cache_grp_sn_sam_cfgN register in the RN SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD28`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-350: cmn_hns_sam_control2

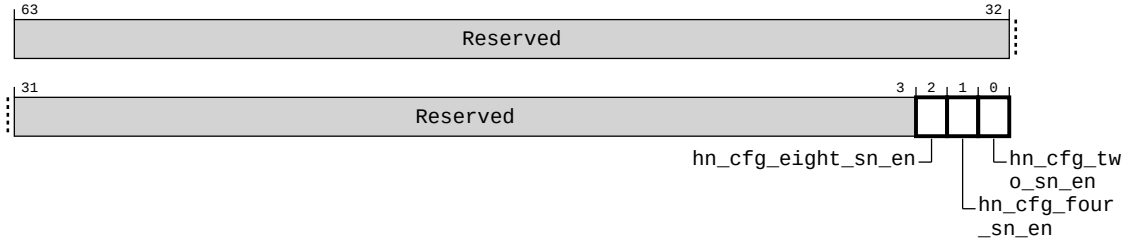


Table 4-366: cmn_hns_sam_control2 attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	hn_cfg_eight_sn_en	Enables 8-SN configuration	RW	1'b0
[1]	hn_cfg_four_sn_en	Enables 4-SN configuration	RW	1'b0
[0]	hn_cfg_two_sn_en	Enables 2-SN configuration	RW	1'b0

4.3.10.70 cmn_hns_sam_memregion0

Configures range-based memory region 0 in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD08`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-351: cmn_hns_sam_memregion0

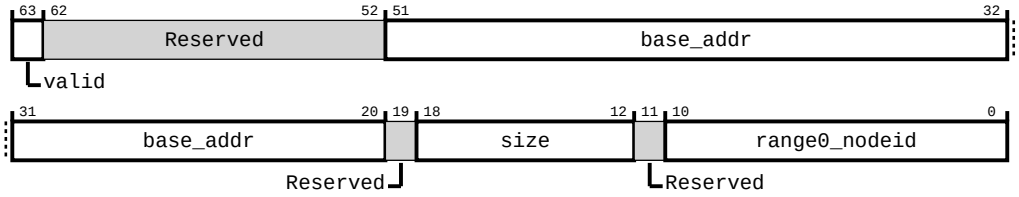


Table 4-367: cmn_hns_sam_memregion0 attributes

Bits	Name	Description	Type	Reset
[63]	valid	Memory region 0 valid **1'b0** Not valid **1'b1** valid for memory region comparison	RW	1'h0
[62:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:20]	base_addr	Base address of memory region 0 CONSTRAINT: Must be an integer multiple of region size.	RW	32'h0
[19]	Reserved	Reserved	RO	-
[18:12]	size	Memory region 0 size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'h0
[11]	Reserved	Reserved	RO	-
[10:0]	range0_nodeid	Memory region 0 target node ID	RW	11'h0

4.3.10.71 cmn_hns_sam_memregion0_end_addr

Configures end address memory region 0 in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD38`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-352: cmn_hns_sam_memregion0_end_addr

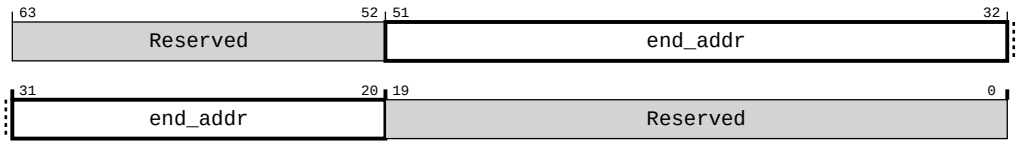


Table 4-368: cmn_hns_sam_memregion0_end_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	end_addr	End address of memory region 0	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

4.3.10.72 cmn_hns_sam_memregion1

Configures range-based memory region 1 in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD10`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-353: cmn_hns_sam_memregion1

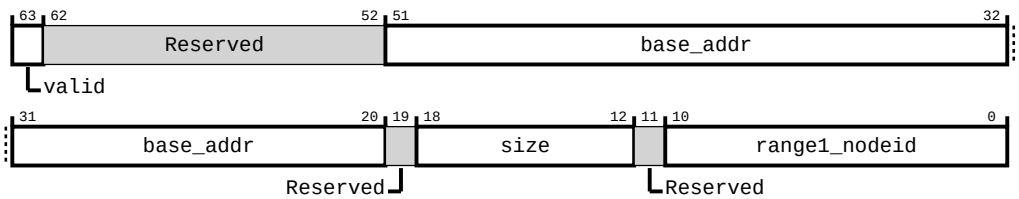


Table 4-369: cmn_hns_sam_memregion1 attributes

Bits	Name	Description	Type	Reset
[63]	valid	Memory region 1 valid **1'b0** Not valid **1'b1** valid for memory region comparison	RW	1'h0
[62:52]	Reserved	Reserved	RO	-
[51:20]	base_addr	Base address of memory region 1 CONSTRAINT: Must be an integer multiple of region size.	RW	32'h0
[19]	Reserved	Reserved	RO	-
[18:12]	size	Memory region 1 size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'h0
[11]	Reserved	Reserved	RO	-
[10:0]	range1_nodeid	Memory region 1 target node ID	RW	11'h0

4.3.10.73 cmn_hns_sam_memregion1_end_addr

Configures end address memory region 1 in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD40`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-354: cmn_hns_sam_memregion1_end_addr

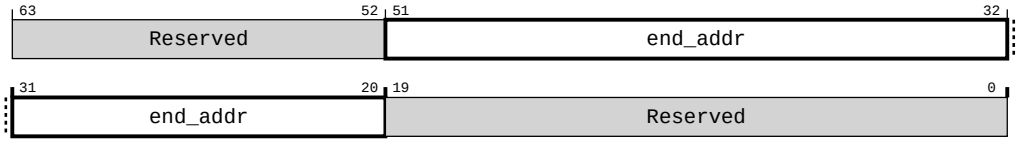


Table 4-370: cmn_hns_sam_memregion1_end_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	end_addr	End address of memory region 1	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

4.3.10.74 cmn_hns_sam_sn_properties

Configures properties for all six SN targets and two range-based SN targets.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD18`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-355: cmn_hns_sam_sn_properties

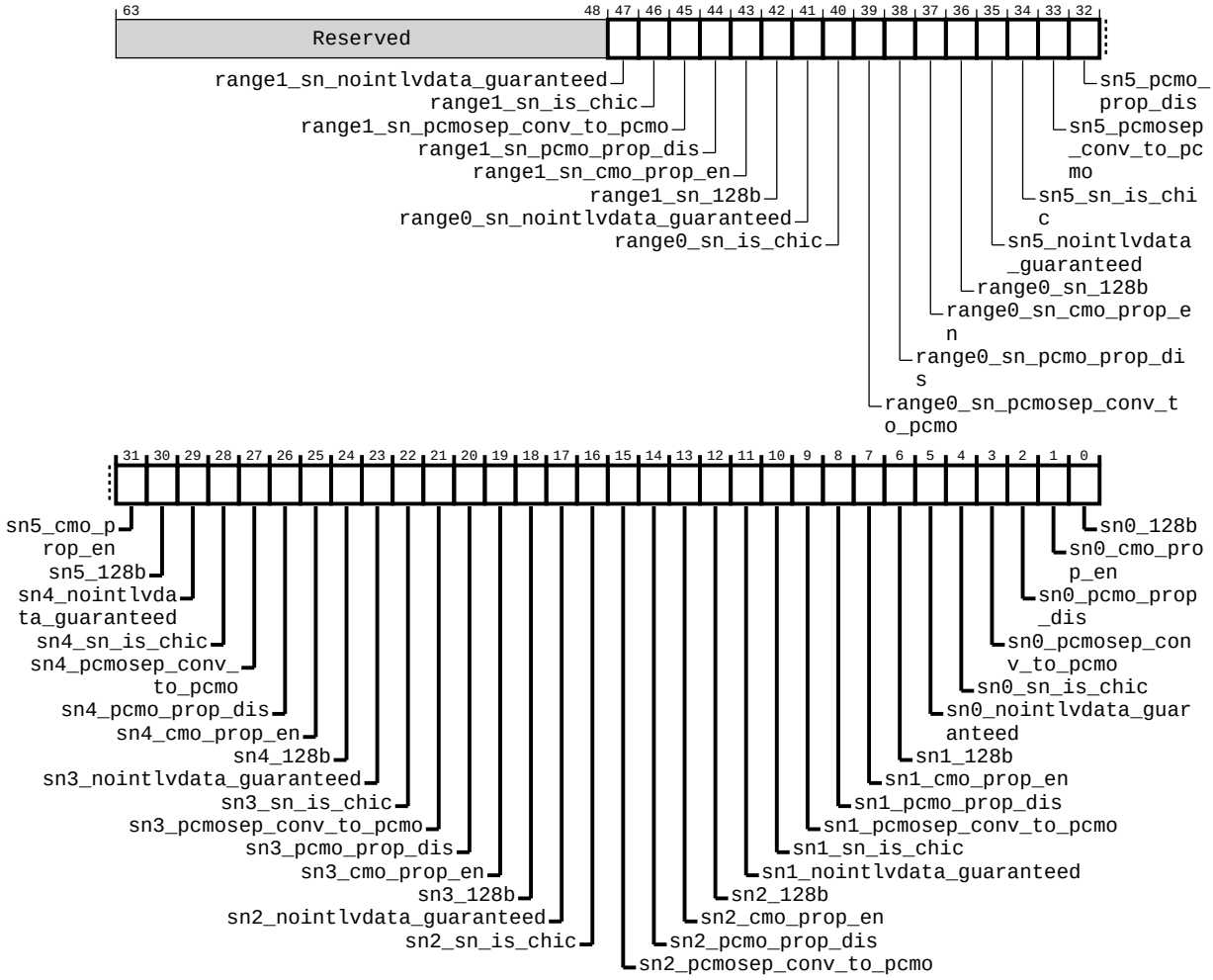


Table 4-371: cmn_hns_sam_sn_properties attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47]	range1_sn_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[46]	range1_sn_is_chic	Indicates that the range 1 SN is a CHI-C SN when set	RW	1'b0
[45]	range1_sn_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for range 1 SN when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[44]	range1_sn_pcmo_prop_dis	Disables PCMO (persistent CMO) propagation for range 1 SN when set	RW	1'b0
[43]	range1_sn_cmo_prop_en	Enables CMO propagation for range 1 SN	RW	1'b0
[42]	range1_sn_128b	Data width of range 1 SN **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[41]	range0_sn_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0

Bits	Name	Description	Type	Reset
[40]	range0_sn_is_chic	Indicates that the range 0 SN is a CHI-C SN when set	RW	1'b0
[39]	range0_sn_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for range 0 SN when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[38]	range0_sn_pcmo_prop_dis	Disables PCMO (persistent CMO) propagation for range 0 SN when set	RW	1'b0
[37]	range0_sn_cmo_prop_en	Enables CMO propagation for range 0 SN	RW	1'b0
[36]	range0_sn_128b	Data width of range 0 SN **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[35]	sn5_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[34]	sn5_sn_is_chic	Indicates that SN5 is a CHI-C SN when set	RW	1'b0
[33]	sn5_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 5 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[32]	sn5_pcmo_prop_dis	Disables PCMO propagation for SN 5 when set	RW	1'b0
[31]	sn5_cmo_prop_en	Enables CMO propagation for SN 5 when set	RW	1'b0
[30]	sn5_128b	Data width of SN 5 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[29]	sn4_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[28]	sn4_sn_is_chic	Indicates that SN4 is a CHI-C SN when set	RW	1'b0
[27]	sn4_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 4 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[26]	sn4_pcmo_prop_dis	Disables PCMO propagation for SN 4 when set	RW	1'b0
[25]	sn4_cmo_prop_en	Enables CMO propagation for SN 4 when set	RW	1'b0
[24]	sn4_128b	Data width of SN 4 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[23]	sn3_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[22]	sn3_sn_is_chic	Indicates that SN3 is a CHI-C SN when set	RW	1'b0
[21]	sn3_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 3 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[20]	sn3_pcmo_prop_dis	Disables PCMO propagation for SN 3 when set	RW	1'b0
[19]	sn3_cmo_prop_en	Enables CMO propagation for SN 3 when set	RW	1'b0
[18]	sn3_128b	Data width of SN 3 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[17]	sn2_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[16]	sn2_sn_is_chic	Indicates that SN2 is a CHI-C SN when set	RW	1'b0
[15]	sn2_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 2 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[14]	sn2_pcmo_prop_dis	Disables PCMO propagation for SN 2 when set	RW	1'b0

Bits	Name	Description	Type	Reset
[13]	sn2_cmo_prop_en	Enables CMO propagation for SN 2 when set	RW	1'b0
[12]	sn2_128b	Data width of SN 2 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[11]	sn1_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[10]	sn1_sn_is_chic	Indicates that SN1 is a CHI-C SN when set	RW	1'b0
[9]	sn1_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 1 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[8]	sn1_pcmo_prop_dis	Disables PCMO propagation for SN 1 when set	RW	1'b0
[7]	sn1_cmo_prop_en	Enables CMO propagation for SN 1 when set	RW	1'b0
[6]	sn1_128b	Data width of SN 1 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[5]	sn0_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[4]	sn0_sn_is_chic	Indicates that SN0 is a CHI-C SN when set	RW	1'b0
[3]	sn0_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 0 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	sn0_pcmo_prop_dis	Disables PCMO propagation for SN 0 when set	RW	1'b0
[1]	sn0_cmo_prop_en	Enables CMO propagation for SN 0 when set	RW	1'b0
[0]	sn0_128b	Data width of SN 0 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0

4.3.10.75 cmn_hns_sam_6sn_nodeid

Configures node IDs for subordinate nodes 3 to 5 in 6-SN configuration mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hD20`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-356: cmn_hns_sam_6sn_nodeid

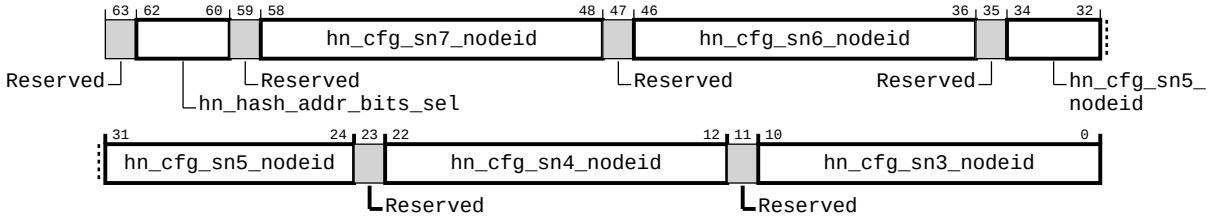


Table 4-372: cmn_hns_sam_6sn_nodeid attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:60]	hn_hash_addr_bits_sel	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[59]	Reserved	Reserved	RO	-
[58:48]	hn_cfg_sn7_nodeid	SN 7 node ID	RW	11'h0
[47]	Reserved	Reserved	RO	-
[46:36]	hn_cfg_sn6_nodeid	SN 6 node ID	RW	11'h0
[35]	Reserved	Reserved	RO	-
[34:24]	hn_cfg_sn5_nodeid	SN 5 node ID	RW	11'h0
[23]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[22:12]	hn_cfg_sn4_nodeid	SN 4 node ID	RW	11'h0
[11]	Reserved	Reserved	RO	-
[10:0]	hn_cfg_sn3_nodeid	SN 3 node ID	RW	11'h0

4.3.10.76 cmn_hns_sam_sn_properties1

Configures additional properties for all six SN targets and two range-based SN targets.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCE8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-357: cmn_hns_sam_sn_properties1

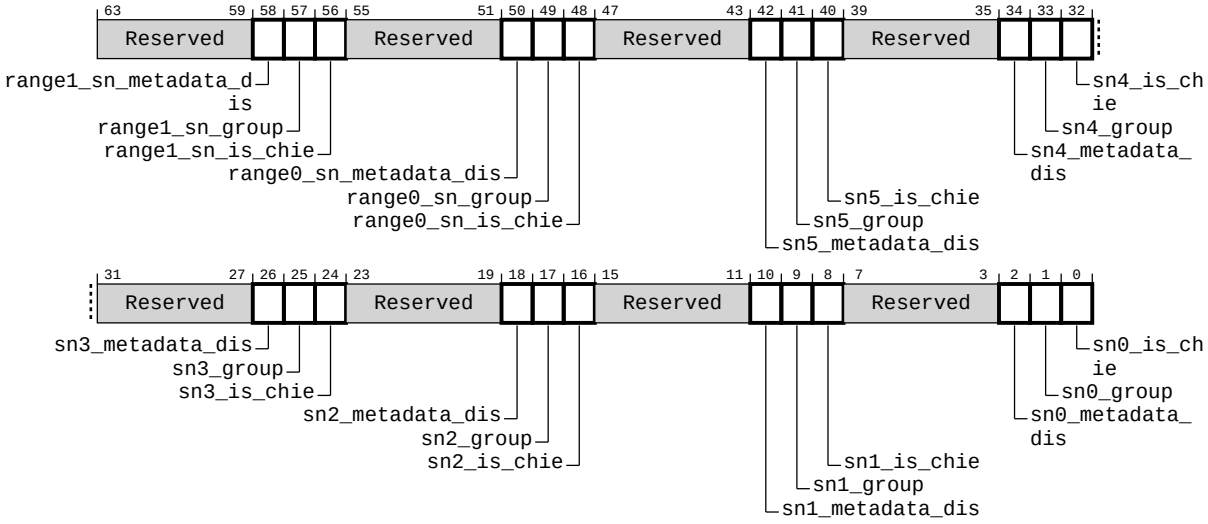


Table 4-373: cmn_hns_sam_sn_properties1 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58]	range1_sn_metadata_dis	HNS implements metadata termination flow for Range 1 SN when set	RW	1'b0
[57]	range1_sn_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[56]	range1_sn_is_chie	Range 1 SN supports CHI-E	RW	1'b0
[55:51]	Reserved	Reserved	RO	-
[50]	range0_sn_metadata_dis	HNS implements metadata termination flow for Range 0 SN when set	RW	1'b0
[49]	range0_sn_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[48]	range0_sn_is_chie	Range 0 SN supports CHI-E	RW	1'b0
[47:43]	Reserved	Reserved	RO	-
[42]	sn5_metadata_dis	HNS implements metadata termination flow for SN 5 when set	RW	1'b0
[41]	sn5_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[40]	sn5_is_chie	SN 5 supports CHI-E	RW	1'b0
[39:35]	Reserved	Reserved	RO	-
[34]	sn4_metadata_dis	HNS implements metadata termination flow for SN 4 when set	RW	1'b0

Bits	Name	Description	Type	Reset
[33]	sn4_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[32]	sn4_is_chie	SN 4 supports CHI-E	RW	1'b0
[31:27]	Reserved	Reserved	RO	-
[26]	sn3_metadata_dis	HNS implements metadata termination flow for SN 3 when set	RW	1'b0
[25]	sn3_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[24]	sn3_is_chie	SN 3 supports CHI-E	RW	1'b0
[23:19]	Reserved	Reserved	RO	-
[18]	sn2_metadata_dis	HNS implements metadata termination flow for SN 2 when set	RW	1'b0
[17]	sn2_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[16]	sn2_is_chie	SN 2 supports CHI-E	RW	1'b0
[15:11]	Reserved	Reserved	RO	-
[10]	sn1_metadata_dis	HNS implements metadata termination flow for SN 1 when set	RW	1'b0
[9]	sn1_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[8]	sn1_is_chie	SN 1 supports CHI-E	RW	1'b0
[7:3]	Reserved	Reserved	RO	-
[2]	sn0_metadata_dis	HNS implements metadata termination flow for SN 0 when set	RW	1'b0
[1]	sn0_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[0]	sn0_is_chie	SN 0 supports CHI-E	RW	1'b0

4.3.10.77 cmn_hns_sam_sn_properties2

Configures properties for SN-7 & SN-8.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD30

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-358: cmn_hns_sam_sn_properties2

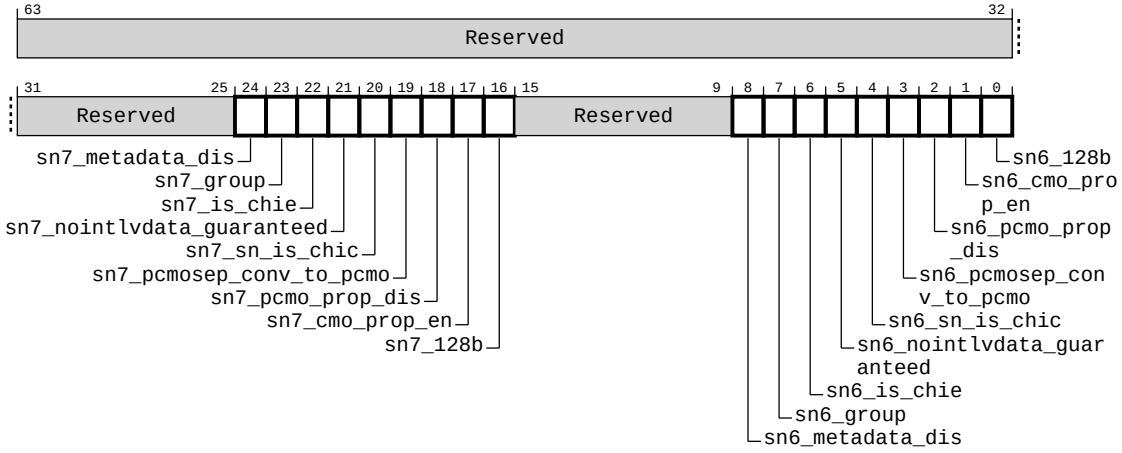


Table 4-374: cmn_hns_sam_sn_properties2 attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	sn7_metadata_dis	HNS implements metadata termination flow for SN 7 when set	RW	1'b0
[23]	sn7_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[22]	sn7_is_chie	SN 7 supports CHI-E	RW	1'b0
[21]	sn7_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[20]	sn7_sn_is_chic	Indicates that SN7 is a CHI-C SN when set	RW	1'b0
[19]	sn7_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 7 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0

Bits	Name	Description	Type	Reset
[18]	sn7_pcmo_prop_dis	Disables PCMO propagation for SN 7 when set	RW	1'b0
[17]	sn7_cmo_prop_en	Enables CMO propagation for SN 7 when set	RW	1'b0
[16]	sn7_128b	Data width of SN 7 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[15:9]	Reserved	Reserved	RO	-
[8]	sn6_metadata_dis	HNS implements metadata termination flow for SN 6 when set	RW	1'b0
[7]	sn6_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[6]	sn6_is_chie	SN 6 supports CHI-E	RW	1'b0
[5]	sn6_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[4]	sn6_sn_is_chic	Indicates that SN6 is a CHI-C SN when set	RW	1'b0
[3]	sn6_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 6 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	sn6_pcmo_prop_dis	Disables PCMO propagation for SN 6 when set	RW	1'b0
[1]	sn6_cmo_prop_en	Enables CMO propagation for SN 6 when set	RW	1'b0
[0]	sn6_128b	Data width of SN 6 **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0

4.3.10.78 cmn_hns_cml_port_aggr_grp0-4_add_mask

There are 5 iterations of this register. The index ranges from 0 to 4. Configures the CCIX port aggregation address mask for group 0.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-4) : 16'hF80 + #{8 * index}
index(5-31) : 16'h6000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-359: cmn_hns_cml_port_aggr_grp0-4_add_mask

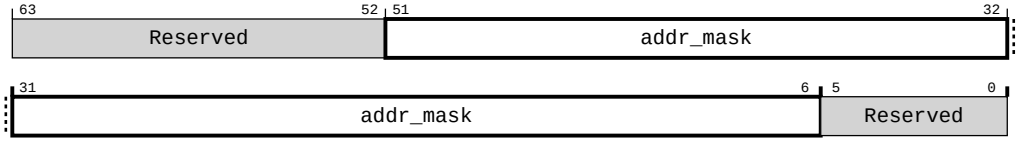


Table 4-375: cmn_hns_cml_port_aggr_grp0-4_add_mask attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address mask to be applied before hashing	RW	46'h3FFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

4.3.10.79 cmn_hns_cml_port_aggr_grp5-31_add_mask

There are 27 iterations of this register. The index ranges from 5 to 31. Configures the CCIX port aggregation address mask for group 0.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-4) : 16'hF80 + #{8 * index}
index(5-31) : 16'h6000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-360: cmn_hns_cml_port_aggr_grp5-31_add_mask

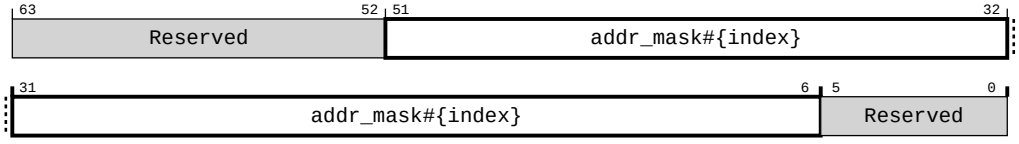


Table 4-376: cmn_hns_cml_port_aggr_grp5-31_add_mask attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask#{index}	Address mask to be applied before hashing	RW	46'h3FFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

4.3.10.80 cmn_hns_cml_port_aggr_grp_reg0-12

There are 13 iterations of this register. The index ranges from 0 to 12. Configures the CCIX port aggregation port Node IDs.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-1) : 16'hFB0 + #{8 * index}
index(2-12) : 16'h6100 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-361: cmn_hns_cml_port_aggr_grp_reg0-12

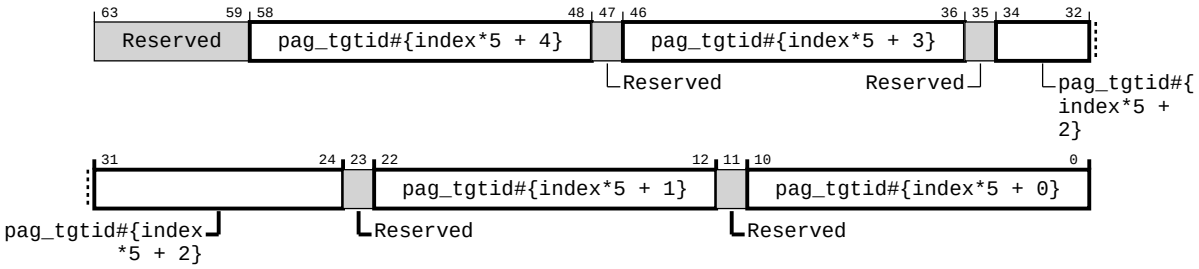


Table 4-377: cmn_hns_cml_port_aggr_grp_reg0-12 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	pag_tgtid#{index*5 + 4}	Specifies the target ID #{index*5 + 4} for CPAG	RW	11'b0
[47]	Reserved	Reserved	RO	-
[46:36]	pag_tgtid#{index*5 + 3}	Specifies the target ID #{index*5 + 3} for CPAG	RW	11'b0
[35]	Reserved	Reserved	RO	-
[34:24]	pag_tgtid#{index*5 + 2}	Specifies the target ID {index*5 + 2} for CPAG	RW	11'b0
[23]	Reserved	Reserved	RO	-
[22:12]	pag_tgtid#{index*5 + 1}	Specifies the target ID {index*5 + 1} for CPAG	RW	11'b0
[11]	Reserved	Reserved	RO	-
[10:0]	pag_tgtid#{index*5 + 0}	Specifies the target ID {index*5 + 0} for CPAG	RW	11'b0

4.3.10.81 cmn_hns_cml_port_aggr_ctrl_reg

Configures the CCIX port aggregation port groups

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hFD0

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-362: cmn_hns_cml_port_aggr_ctrl_reg

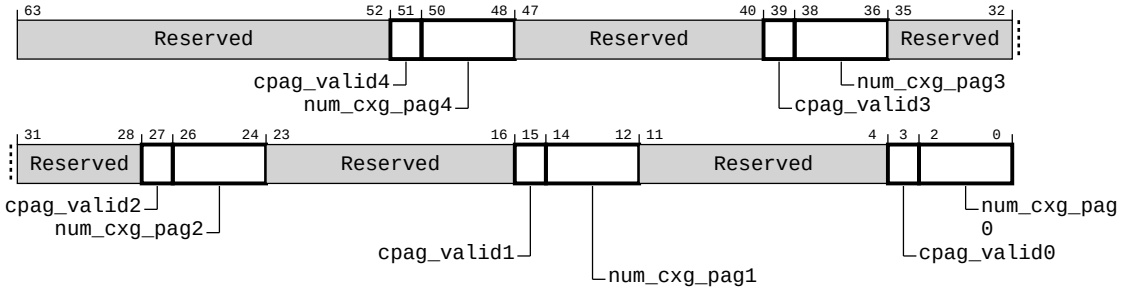


Table 4-378: cmn_hns_cml_port_aggr_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	cpag_valid4	Valid programming for CPAG4, Enabled by default (backward compatible)	RW	1'b1
[50:48]	num_cxg_pag4	Specifies the number of CXRAs in CPAG4 Constraint: May use pag_tgtid8 through pag_tgtid9 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid3	Valid programming for CPAG + 3}, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[38:36]	num_cxg_pag3	Specifies the number of CXRAs in CPAG3 Constraint: May use pag_tgtid6 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[35:28]	Reserved	Reserved	RO	-
[27]	cpag_valid2	Valid programming for CPAG + 2}, Enabled by default (backward compatible)	RW	1'b1
[26:24]	num_cxg_pag2	Specifies the number of CXRAs in CPAG2 Constraint: May use pag_tgtid4 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid1	Valid programming for CPAG + 1}, Enabled by default (backward compatible)	RW	1'b1
[14:12]	num_cxg_pag1	Specifies the number of CXRAs in CPAG1 Constraint: May use pag_tgtid2 through pag_tgtid3 of cmn_hns_cml_port_aggr_grp_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid0	Valid programming for CPAG, Enabled by default (backward compatible)	RW	1'b1
[2:0]	num_cxg_pag0	Specifies the number of CXRAs in CPAG0 Constraint: May use pag_tgtid0 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000

4.3.10.82 cmn_hns_cml_port_aggr_ctrl_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures the CCIX port aggregation port groups

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(1-6) : 16'h6200 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-363: cmn_hns_cml_port_aggr_ctrl_reg1-6

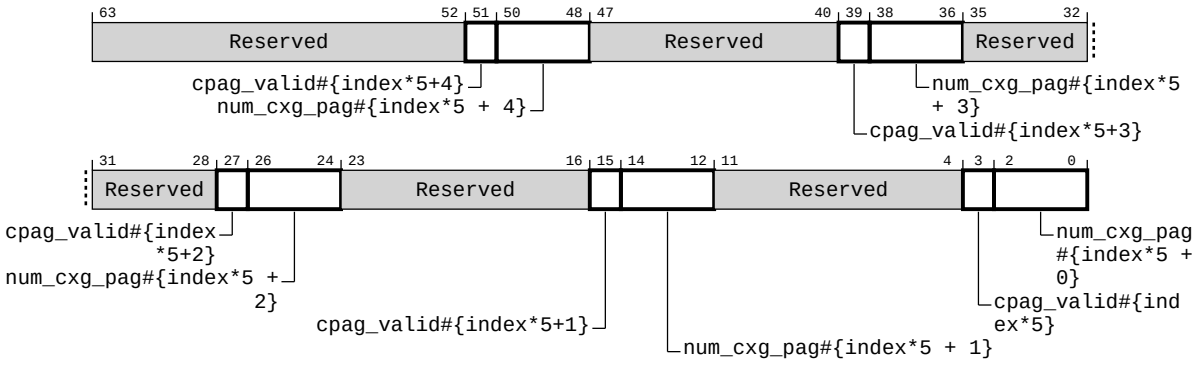


Table 4-379: cmn_hns_cml_port_aggr_ctrl_reg1-6 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51]	cpag_valid#{index*5+4}	Valid programming for CPAG #{index*5 + 4}, Enabled by default (backward compatible)	RW	1'b1
[50:48]	num_cxg_pag#{index*5 + 4}	Specifies the number of CXRAs in CPAG4#{index*5 + 4} Constraint: May use pag_tgtid8 through pag_tgtid9 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid#{index*5+3}	Valid programming for CPAG #{index*5 + 3}, Enabled by default (backward compatible)	RW	1'b1
[38:36]	num_cxg_pag#{index*5 + 3}	Specifies the number of CXRAs in CPAG3#{index*5 + 3} Constraint: May use pag_tgtid6 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[35:28]	Reserved	Reserved	RO	-
[27]	cpag_valid#{index*5+2}	Valid programming for CPAG #{index*5 + 2}, Enabled by default (backward compatible)	RW	1'b1
[26:24]	num_cxg_pag#{index*5 + 2}	Specifies the number of CXRAs in CPAG2#{index*5 + 2} Constraint: May use pag_tgtid4 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid#{index*5+1}	Valid programming for CPAG #{index*5 + 1}, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[14:12]	num_cxg_pag#{index*5 + 1}	Specifies the number of CXRAs in CPAG1#{index*5 + 1} Constraint: May use pag_tgtid2 through pag_tgtid3 of cmn_hns_cml_port_aggr_grp_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid#{index*5}	Valid programming for CPAG #{index*5}, Enabled by default (backward compatible)	RW	1'b1
[2:0]	num_cxg_pag#{index*5 + 0}	Specifies the number of CXRAs in CPAG#{index*5 + 0} Constraint: May use pag_tgtid0 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 **3'b000** 1 port used **3'b001** 2 ports used **3'b010** 4 ports used **3'b011** 8 ports used **3'b100** 16 ports used **3'b101** 32 ports used **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b000

4.3.10.83 cmn_hns_abf_lo_addr

Lower address range for Address Based Flush (ABF) [51:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF50`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-364: cmn_hns_abf_lo_addr

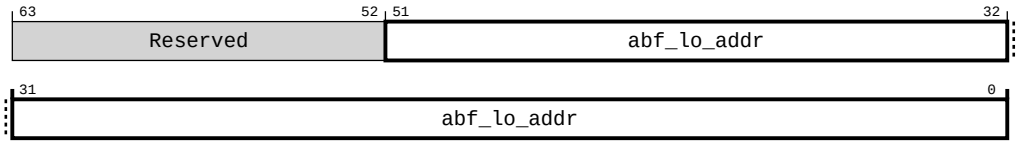


Table 4-380: cmn_hns_abf_lo_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:0]	abf_lo_addr	Lower address range for ABF	RW	52'b0

4.3.10.84 cmn_hns_abf_hi_addr

Upper address range for Address Based Flush (ABF) [51:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF58`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-365: cmn_hns_abf_hi_addr

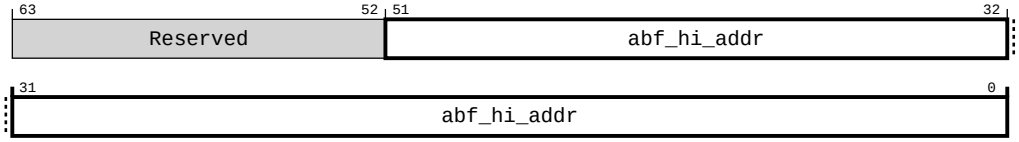


Table 4-381: cmn_hns_abf_hi_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:0]	abf_hi_addr	Upper address range for ABF	RW	52'b0

4.3.10.85 cmn_hns_abf_pr

Functions as the Address Based Flush (ABF) policy register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF60`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-366: cmn_hns_abf_pr

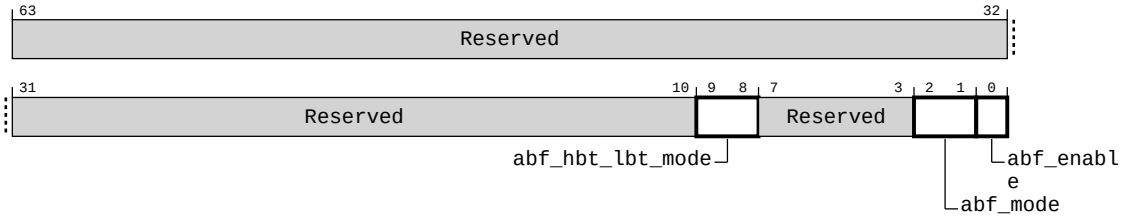


Table 4-382: cmn_hns_abf_pr attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:8]	abf_hbt_lbt_mode	ABF HBT/LBT Flush mode **2'b00** All addresses in ABF range (HBT and LBT) flushed **2'b01** All HBT addresses in ABF range flushed **2'b10** All LBT addresses in ABF range flushed **2'b11** Reserved	RW	2'b00
[7:3]	Reserved	Reserved	RO	-
[2:1]	abf_mode	ABF mode **2'b00** Clean Invalidate; WB dirty data and invalidate local copy **2'b01** Make Invalidate; invalidate without writing back dirty data **2'b10** Clean Shared; WB dirty data and can keep clean copy **2'b11** Reserved	RW	2'b00
[0]	abf_enable	Start Address Based Flushing based on high and low address ranges	RW	1'b0

4.3.10.86 cmn_hns_abf_sr

Functions as the Address Based Flush (ABF) status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hF68`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-367: cmn_hns_abf_sr

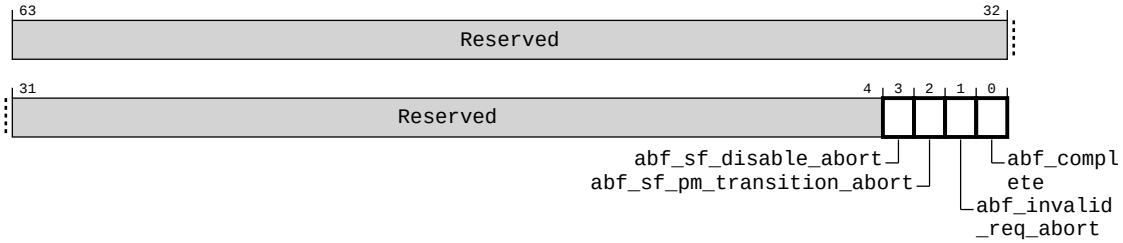


Table 4-383: cmn_hns_abf_sr attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	abf_sf_disable_abort	ABF aborted due to SF not being enabled, either by configuration or double-bit ECC error	RO	1'b0
[2]	abf_sf_pm_transition_abort	ABF aborted due to PM transition while ABF in progress, or both PM and ABF requested at the same time	RO	1'b0
[1]	abf_invalid_req_abort	ABF request made while PM is not in FAM/HAM/SF_ONLY mode; request aborted in this case	RO	1'b0
[0]	abf_complete	ABF completed	RO	1'b0

4.3.10.87 cmn_hns_cbusy_write_limit_ctl

Cbusy threshold limits for POCQ write entries. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1000`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-368: cmn_hns_cbusy_write_limit_ctl

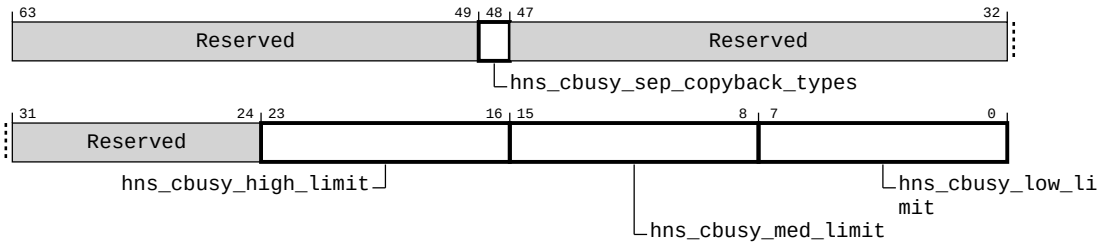


Table 4-384: cmn_hns_cbusy_write_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:49]	Reserved	Reserved	RO	-
[48]	hns_cbusy_sep_copyback_types	Enables copyback and non-copyback write type separation in cbusy calculation	RW	1'b0
[47:24]	Reserved	Reserved	RO	-
[23:16]	hns_cbusy_high_limit	POCQ limit for Write CBusy High	RW	Configuration dependent
[15:8]	hns_cbusy_med_limit	POCQ limit for Write CBusy Med	RW	Configuration dependent
[7:0]	hns_cbusy_low_limit	POCQ limit for Write CBusy Low	RW	Configuration dependent

4.3.10.88 cmn_hns_cbusy_resp_ctl

Contols the responses sent from HNS to RNF. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1008

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-369: cmn_hns_cbusy_resp_ctl

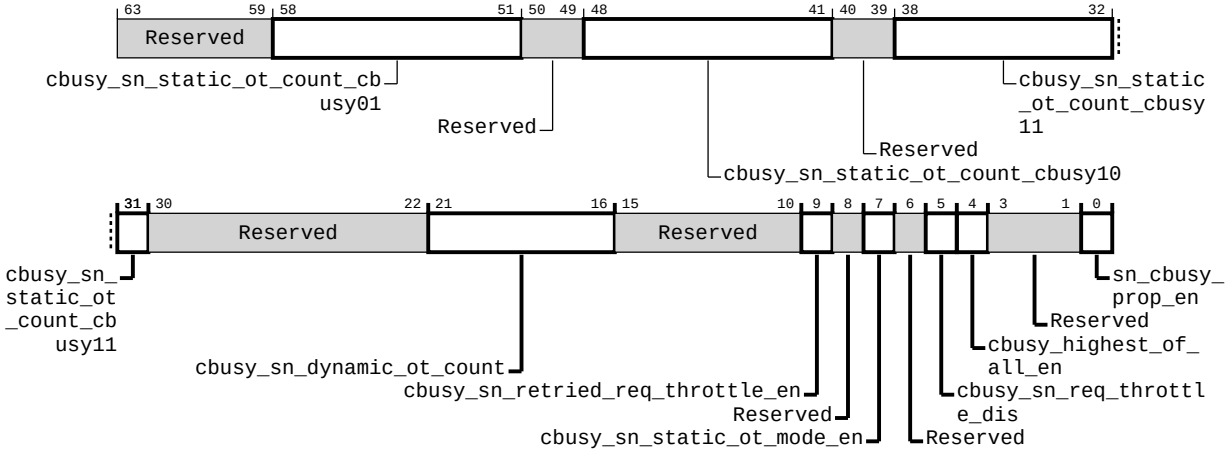


Table 4-385: cmn_hns_cbusy_resp_ctl attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:51]	cbusy_sn_static_ot_count_cbusy01	Specifies the maximum number of transactions to SN-F when SN Cbusy=01 in static throttling mode. CONSTRAINT: Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1	RW	Configuration dependent
[50:49]	Reserved	Reserved	RO	-
[48:41]	cbusy_sn_static_ot_count_cbusy10	Specifies the maximum number of transactions to SN-F when SN Cbusy=10 in static throttling mode. CONSTRAINT: Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1 and less than cbusy_sn_static_ot_count_cbusy01	RW	Configuration dependent
[40:39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:31]	cbusy_sn_static_ot_count_cbusy11	Specifies the maximum number of transactions to SN-F when SN Cbusy=11 in static throttling mode. CONSTRAINT: Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1 and less than cbusy_sn_static_ot_count_cbusy10	RW	Configuration dependent
[30:22]	Reserved	Reserved	RO	-
[21:16]	cbusy_sn_dynamic_ot_count	Specifies the granularity at which HN-F will dynamically throttle transactions to SN-F. CONSTRAINT: 1,2,4,8 are the the allowed values	RW	6'b000100
[15:10]	Reserved	Reserved	RO	-
[9]	cbusy_sn_retried_req_throttle_en	Enables throttling retried requests with static grants (from SN) along with dynamic credit requests	RW	1'b0
[8]	Reserved	Reserved	RO	-
[7]	cbusy_sn_static_ot_mode_en	Controls cbusy between HN-F and SN-F **1'b0** HN-F will dynamically throttle outstanding requests to SN-F **1'b1** HN-F will use fixed transactions count at each CBusy level at 1/4th POCQ granularity CONSTRAINT: For SN request throttling, CBusy aggregation is always based on SN_CBusy[1:0] and cbusy_alt_mode_en is inapplicable	RW	1'b0
[6]	Reserved	Reserved	RO	-
[5]	cbusy_sn_req_throttle_dis	Disables Cbusy based request throttling from HNS to SNF when set to 1'b1	RW	1'b0
[4]	cbusy_highest_of_all_en	Controls cbusy between HN-F and SN-F **1'b0** Will send the HN-F or SN-F as configured **1'b1** Will select highest CBusy value between the SN-F and HN-F	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	sn_cbusy_prop_en	Controls HN-F and SN-F cbusy on responses to RN-F **1'b0** HN-F's POCQ Cbusy is sent **1'b1** SN-F's Cbusy is sent	RW	1'b0

4.3.10.89 cmn_hns_cbusy_sn_ctl

Controls the SN-F cbusy thresholds. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1010

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-370: cmn_hns_cbusy_sn_ctl

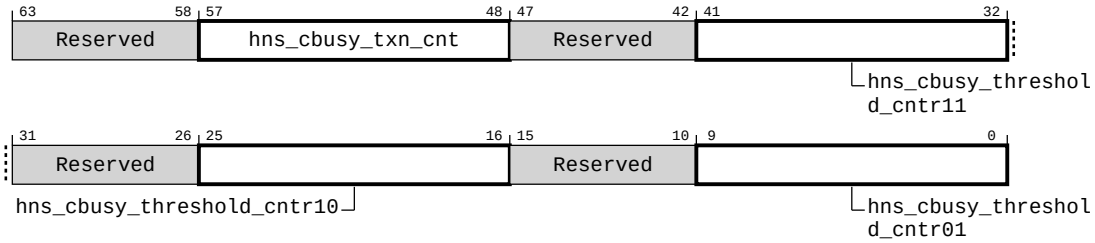


Table 4-386: cmn_hns_cbusy_sn_ctl attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	hns_cbusy_txn_cnt	Number of transactions over which the counters are tracked	RW	10'b0100000000
[47:42]	Reserved	Reserved	RO	-
[41:32]	hns_cbusy_threshold_cntr11	CBusy threshold at which SN-F is considered busy for Counter_11	RW	10'b0000010000
[31:26]	Reserved	Reserved	RO	-
[25:16]	hns_cbusy_threshold_cntr10	CBusy threshold at which SN-F is considered busy for Counter_10	RW	10'b0000100000
[15:10]	Reserved	Reserved	RO	-
[9:0]	hns_cbusy_threshold_cntr01	CBusy threshold at which SN-F is considered busy for Counter_01	RW	10'b0001000000

4.3.10.90 cmn_hns_lbt_cbusy_ctl

Controls the CBusy response for LCN Bound Transactions. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1018`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-371: cmn_hns_lbt_cbusy_ctl

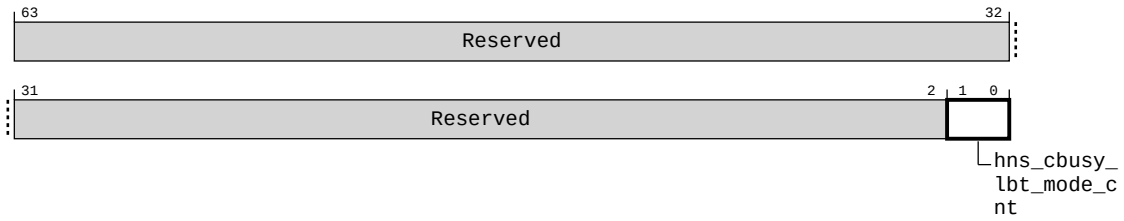


Table 4-387: cmn_hns_lbt_cbusy_ctl attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	hns_cbusy_lbt_mode_cnt	Controls the propagation of Cbusy field for LCN bound transactions. **2'b00** Send HNS POCQ Cbusy on all responses based on the limits programmed in cmn_hns_cbusy_limit_ctl **2'b01** Pass through HNF CBusy on late completion responses (CompData, Comp) **2'b10** Greater of POCQ Cbusy or HNF Cbusy. Applicable to responses where remote Cbusy can be sent	RW	2'b00

4.3.10.91 cmn_hns_pocq_alloc_class_dedicated

Controls Dedicated entries in POCQ for each class.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1020`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-372: cmn_hns_pocq_alloc_class_dedicated

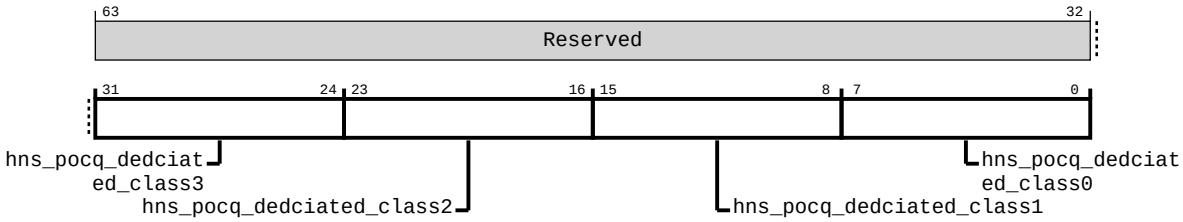


Table 4-388: cmn_hns_pocq_alloc_class_dedicated attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	hns_pocq_dedciated_class3	Dedicated number of entries for Class 3 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class3 < hns_pocq_max_allowed_class3	RW	8'b00000000

Bits	Name	Description	Type	Reset
[23:16]	hns_pocq_dedciated_class2	Dedicated number of entries for Class 2 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class2 < hns_pocq_max_allowed_class2	RW	8'b00000000
[15:8]	hns_pocq_dedciated_class1	Dedicated number of entries for Class 1 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class1 < hns_pocq_max_allowed_class1	RW	8'b00000000
[7:0]	hns_pocq_dedciated_class0	Dedicated number of entries for Class 0 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class0 < hns_pocq_max_allowed_class0	RW	8'b00000000

4.3.10.92 cmn_hns_pocq_alloc_class_max_allowed

Controls Maximum allowed entries in POCQ for each class.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1028`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-373: cmn_hns_pocq_alloc_class_max_allowed

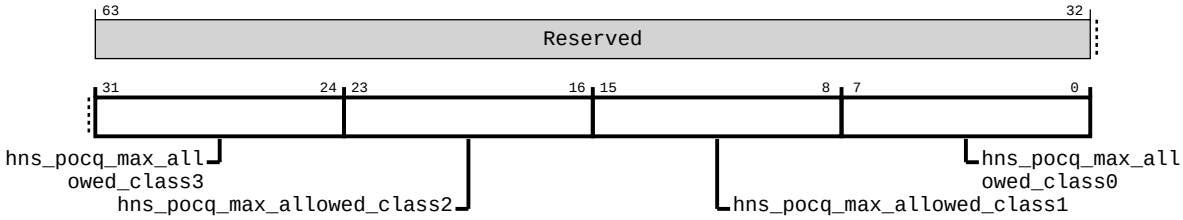


Table 4-389: cmn_hns_pocq_alloc_class_max_allowed attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	hns_pocq_max_allowed_class3	Maximum number of entries for Class 3 in POCQ CONSTRAINT: hns_pocq_dedciated_class3 < hns_pocq_max_allowed_class3	RW	Configuration dependent
[23:16]	hns_pocq_max_allowed_class2	Maximum number of entries for Class 2 in POCQ CONSTRAINT: hns_pocq_dedciated_class2 < hns_pocq_max_allowed_class2	RW	Configuration dependent
[15:8]	hns_pocq_max_allowed_class1	Maximum number of entries for Class 1 in POCQ CONSTRAINT: hns_pocq_dedciated_class1 < hns_pocq_max_allowed_class1	RW	Configuration dependent
[7:0]	hns_pocq_max_allowed_class0	Maximum number of entries for Class 0 in POCQ CONSTRAINT: hns_pocq_dedciated_class0 < hns_pocq_max_allowed_class0	RW	Configuration dependent

4.3.10.93 cmn_hns_pocq_alloc_class_contended_min

Controls Contended minimum entries in POCQ for each class.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1030`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-374: cmn_hns_pocq_alloc_class_contended_min

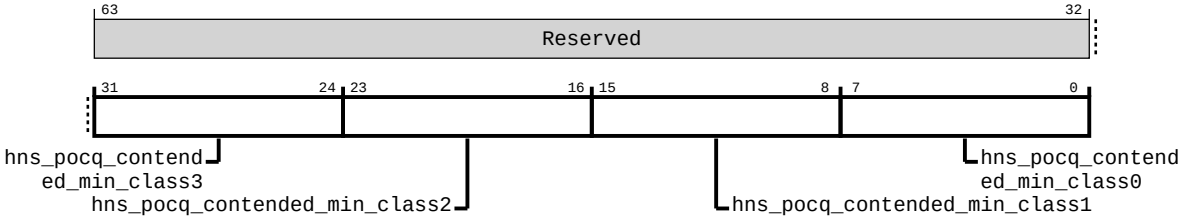


Table 4-390: cmn_hns_pocq_alloc_class_contended_min attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	hns_pocq_contended_min_class3	Contended min entries for Class 3 in POCQ	RW	Configuration dependent
[23:16]	hns_pocq_contended_min_class2	Contended min entries for Class 2 in POCQ	RW	Configuration dependent
[15:8]	hns_pocq_contended_min_class1	Contended min entries for Class 1 in POCQ	RW	Configuration dependent
[7:0]	hns_pocq_contended_min_class0	Contended min entries for Class 0 in POCQ	RW	Configuration dependent

4.3.10.94 cmn_hns_pocq_alloc_misc_max_allowed

Controls Maximum allowed entries in POCQ for SNP, SEQ, and other misc req.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1038`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-375: cmn_hns_pocq_alloc_misc_max_allowed

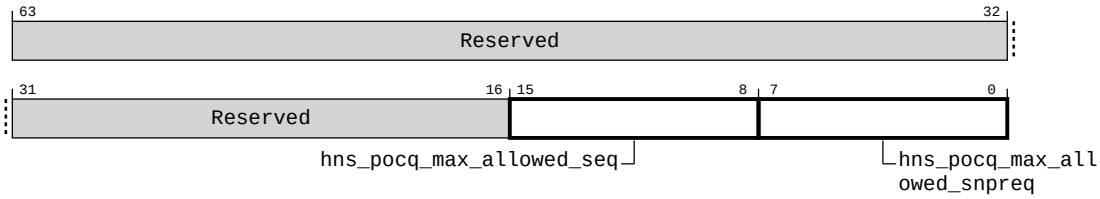


Table 4-391: cmn_hns_pocq_alloc_misc_max_allowed attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	hns_pocq_max_allowed_seq	Maximum number of entries for SEQ in POCQ. Constraint: Only values of 1 or 2 supported.	RW	8'h02
[7:0]	hns_pocq_max_allowed_snpreq	Maximum number of entries for RXSNP requests in POCQ	RW	8'h04

4.3.10.95 cmn_hns_class_ctl

Class misc controls.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1040`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-376: cmn_hns_class_ctl

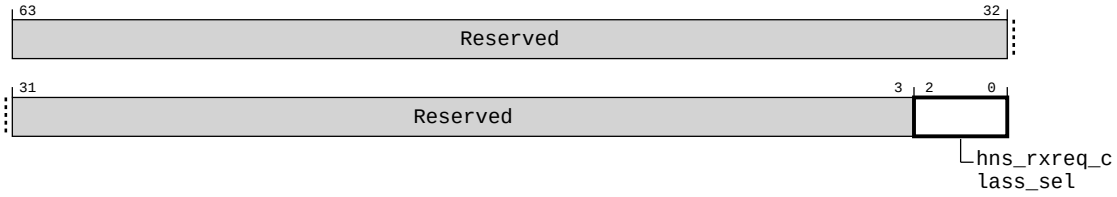


Table 4-392: cmn_hns_class_ctl attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2:0]	hns_rxreq_class_sel	RxReq Class select: **3'b000** QoS based class selection **3'b001** Request Opcode based class selection If un-supported value is programmed, default selection of QoS based is chosen.	RW	3'b000

4.3.10.96 cmn_hns_pocq_qos_class_ctl

QoS bases class identification controls.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1048`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-377: cmn_hns_pocq_qos_class_ctl

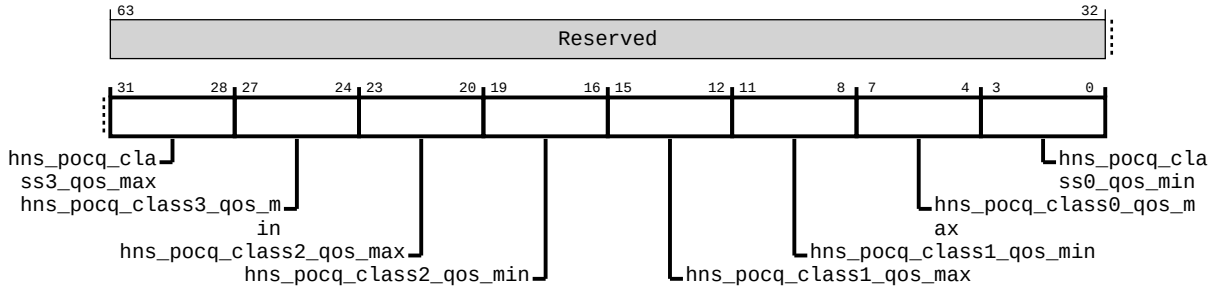


Table 4-393: cmn_hns_pocq_qos_class_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	hns_pocq_class3_qos_max	QoS maximum value for Class 3	RW	4'b0111
[27:24]	hns_pocq_class3_qos_min	QoS minimum value for Class 3	RW	4'b0000
[23:20]	hns_pocq_class2_qos_max	QoS maximum value for Class 2	RW	4'b1011
[19:16]	hns_pocq_class2_qos_min	QoS minimum value for Class 2	RW	4'b1000
[15:12]	hns_pocq_class1_qos_max	QoS maximum value for Class 1	RW	4'b1110
[11:8]	hns_pocq_class1_qos_min	QoS minimum value for Class 1	RW	4'b1100
[7:4]	hns_pocq_class0_qos_max	QoS maximum value for Class 0	RW	4'b1111
[3:0]	hns_pocq_class0_qos_min	QoS minimum value for Class 0	RW	4'b1111

4.3.10.97 cmn_hns_class_pocq_arb_weight_ctl

Per Class weight controls for scheduling requests from POCQ.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1050`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-378: cmn_hns_class_pocq_arb_weight_ctl

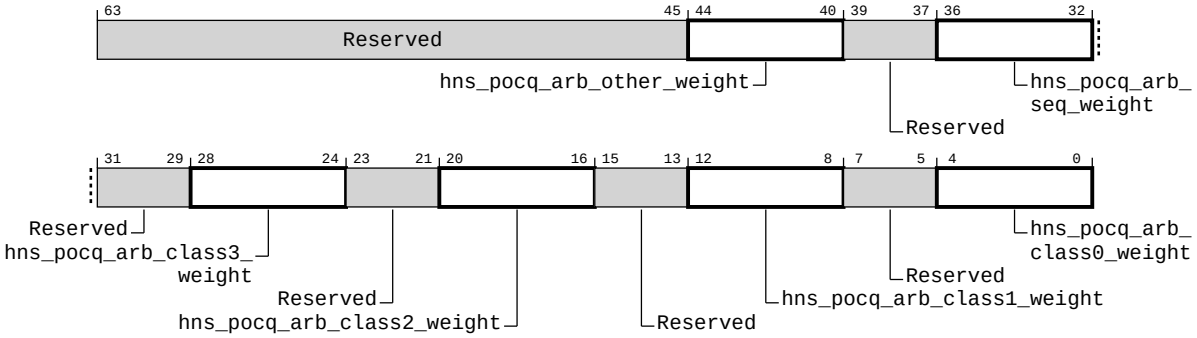


Table 4-394: cmn_hns_class_pocq_arb_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:40]	hns_pocq_arb_other_weight	Other req weight for scheduling requests from POCQ	RW	5'b00000
[39:37]	Reserved	Reserved	RO	-
[36:32]	hns_pocq_arb_seq_weight	SEQ weight for scheduling requests from POCQ	RW	5'b00000
[31:29]	Reserved	Reserved	RO	-
[28:24]	hns_pocq_arb_class3_weight	Class3 weight for scheduling requests from POCQ	RW	5'b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	hns_pocq_arb_class2_weight	Class2 weight for scheduling requests from POCQ	RW	5'b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	hns_pocq_arb_class1_weight	Class1 weight for scheduling requests from POCQ	RW	5'b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	hns_pocq_arb_class0_weight	Class0 weight for scheduling requests from POCQ	RW	5'b00000

4.3.10.98 cmn_hns_class_retry_weight_ctl

Per Class weight controls for Retry Credit grant.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1058`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-379: cmn_hns_class_retry_weight_ctl

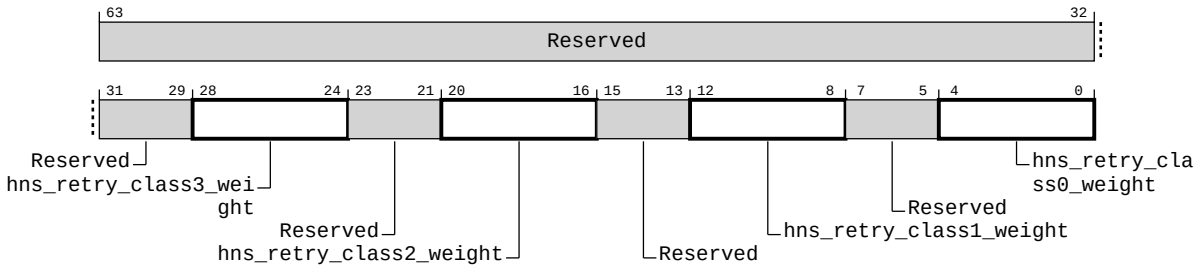


Table 4-395: cmn_hns_class_retry_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:24]	hns_retry_class3_weight	Overall Class3 weight for credit grant arbitration	RW	5'b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	hns_retry_class2_weight	Overall Class2 weight for credit grant arbitration	RW	5'b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	hns_retry_class1_weight	Overall Class1 weight for credit grant arbitration	RW	5'b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	hns_retry_class0_weight	Overall Class0 weight for credit grant arbitration	RW	5'b00000

4.3.10.99 cmn_hns_pocq_misc_retry_weight_ctl

Weight controls for Snoop, SEQ, Flush and other misc POCQ requests.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1060

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-380: cmn_hns_pocq_misc_retry_weight_ctl

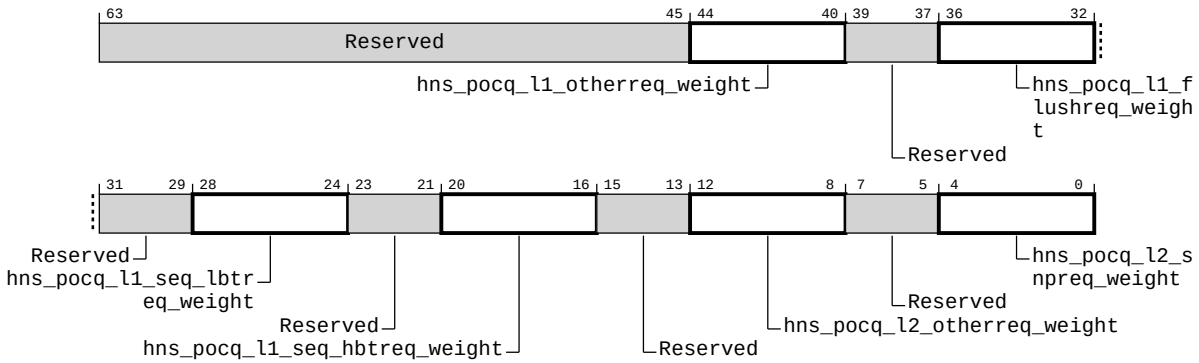


Table 4-396: cmn_hns_pocq_misc_retry_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:40]	hns_pocq_l1_otherreq_weight	Weight for other requests (Ex: Debug Read) for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000

Bits	Name	Description	Type	Reset
[39:37]	Reserved	Reserved	RO	-
[36:32]	hns_pocq_l1_flushreq_weight	Weight for SLF/SF Flush requests for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000
[31:29]	Reserved	Reserved	RO	-
[28:24]	hns_pocq_l1_seq_lbtreq_weight	Weight for SEQ-LBT requests for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	hns_pocq_l1_seq_hbtreq_weight	Weight for SEQ-HBT requests for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	hns_pocq_l2_otherreq_weight	Weight for other requests (Ex: Level 1 arb req) for POCQ allocation arbitration for Level 2. Note: This is second level arb weight control. First level is seq, flush, dbgrd, etc.	RW	5'b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	hns_pocq_l2_snpreq_weight	Weight for external snoop requests for POCQ allocation arbitration for Level 2. Note: This is second level arb weight control. First level is seq, flush, dbgrd, etc	RW	5'b00000

4.3.10.100 cmn_hns_partner_scratch_reg0

Partner scratch register 0

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFE0`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.partner_scratch_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-381: cmn_hns_partner_scratch_reg0

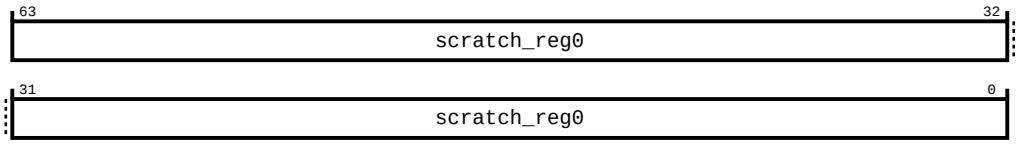


Table 4-397: cmn_hns_partner_scratch_reg0 attributes

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg0	64 bit scratch register 0 wirh read/write access	RW	64'h00000000

4.3.10.101 cmn_hns_partner_scratch_reg1

Partner scratch register 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hFE8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.partner_scratch_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-382: cmn_hns_partner_scratch_reg1

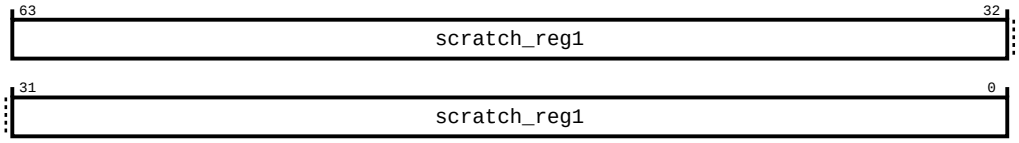


Table 4-398: cmn_hns_partner_scratch_reg1 attributes

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg1	64 bit scratch register 1 wirh read/write access	RW	64'h00000000

4.3.10.102 cmn_hns_cfg_slcsf_dbgrd

Controls access modes for SLC tasg, SLC data, and SF tag debug read.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hB80`

Type

WO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-383: cmn_hns_cfg_slcsf_dbgrd

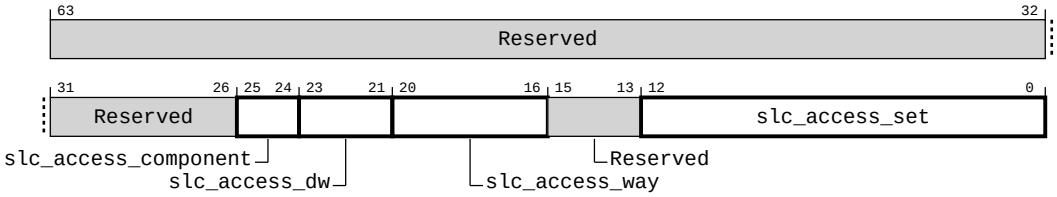


Table 4-399: cmn_hns_cfg_slcsf_dbgrd attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:24]	slc_access_component	Specifies SLC/SF array debug read **2'b01** SLC data read **2'b10** SLC tag read **2'b11** SF tag read	WO	2'b00
[23:21]	slc_access_dw	64-bit chunk address for SLC data debug read access	WO	3'h0
[20:16]	slc_access_way	Way address for SLC/SF debug read access	WO	5'h00
[15:13]	Reserved	Reserved	RO	-
[12:0]	slc_access_set	Set address for SLC/SF debug read access	WO	13'h0

4.3.10.103 cmn_hns_slc_cache_access_slc_tag

Contains SLC tag debug read data bits [63:0]

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hB88`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-384: cmn_hns_slc_cache_access_slc_tag

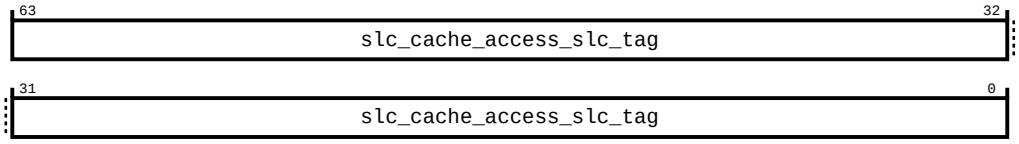


Table 4-400: cmn_hns_slc_cache_access_slc_tag attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_tag	SLC tag debug read data	RO	64'h0

4.3.10.104 cmn_hns_slc_cache_access_slc_tag1

Contains SLC tag debug read data bits [127:64] when present

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hB90

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-385: cmn_hns_slc_cache_access_slc_tag1

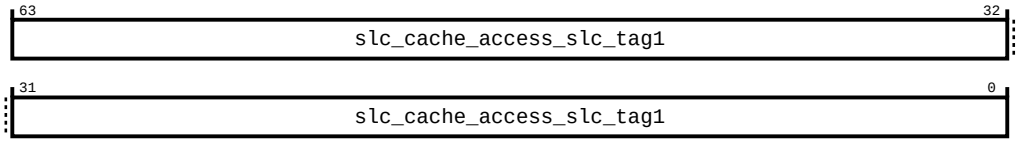


Table 4-401: cmn_hns_slc_cache_access_slc_tag1 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_tag1	SLC tag debug read data	RO	64'h0

4.3.10.105 cmn_hns_slc_cache_access_slc_data

Contains SLC data RAM debug read data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hB98`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-386: cmn_hns_slc_cache_access_slc_data

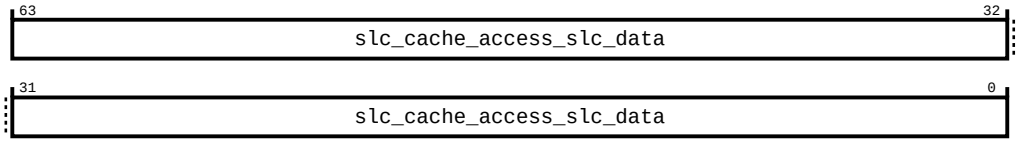


Table 4-402: cmn_hns_slc_cache_access_slc_data attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_data	SLC data RAM debug read data	RO	64'h0

4.3.10.106 cmn_hns_slc_cache_access_slc_mte_tag

Contains MTE Tag data for the corresponding SLC data RAM debug read.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hBC0`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-387: cmn_hns_slc_cache_access_slc_mte_tag

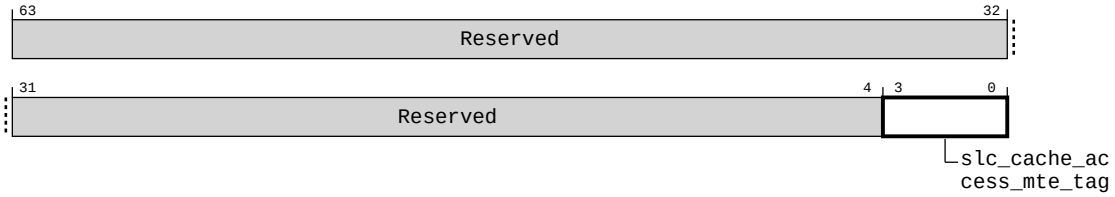


Table 4-403: cmn_hns_slc_cache_access_slc_mte_tag attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	slc_cache_access_mte_tag	SLC MTE TAG corresponding to data RAM debug read data (128bit chunk of data)	RO	4'h0

4.3.10.107 cmn_hns_slc_cache_access_sf_tag

Contains SF tag debug read data. Bits[63:0]

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hBA0`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-388: cmn_hns_slc_cache_access_sf_tag

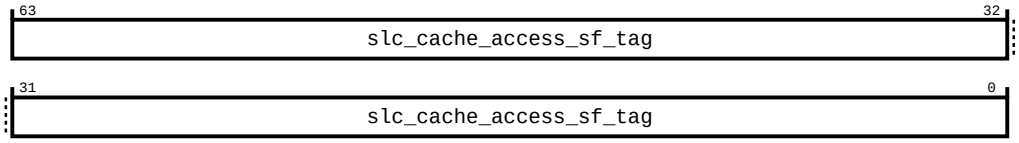


Table 4-404: cmn_hns_slc_cache_access_sf_tag attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag	SF tag debug read data	RO	64'h0

4.3.10.108 cmn_hns_slc_cache_access_sf_tag1

Contains SF tag debug read data bits [127:64], when present in SF Tag

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hBA8`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-389: cmn_hns_slc_cache_access_sf_tag1

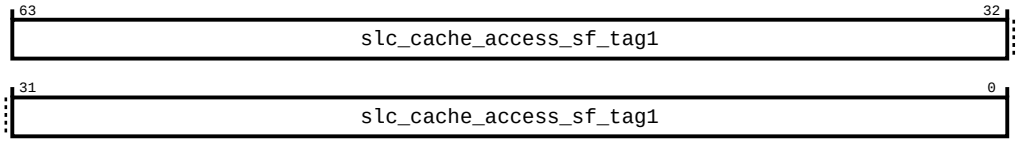


Table 4-405: cmn_hns_slc_cache_access_sf_tag1 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag1	SF tag debug read data	RO	64'h0

4.3.10.109 cmn_hns_slc_cache_access_sf_tag2

Contains SF tag debug read data bits [128:191], when present in SF Tag

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hBB0`

Type

RO

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-390: cmn_hns_slc_cache_access_sf_tag2

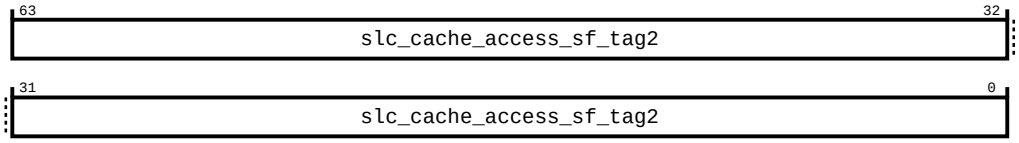


Table 4-406: cmn_hns_slc_cache_access_sf_tag2 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag2	SF tag debug read data	RO	64'h0

4.3.10.110 cmn_hns_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-391: cmn_hns_pmu_event_sel

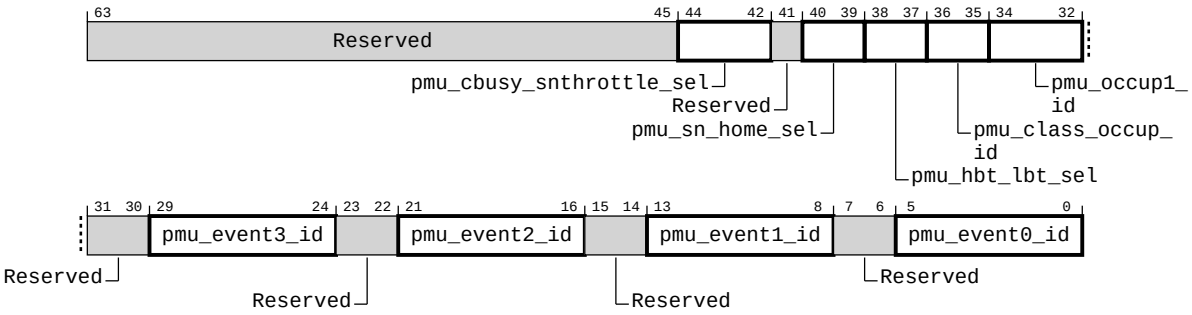


Table 4-407: cmn_hns_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:42]	pmu_cbusy_snthrottle_sel	Filter for selecting specific SN throttle type **3'b000** All SN types throttled **3'b001** SN Group 0 Reads **3'b010** SN Group 0 Non-Reads **3'b011** SN Group 1 Reads **3'b100** SN Group 1 Non-Reads **3'b101** All SN Reads **3'b110** All SN Non-Reads	RW	3'h0
[41]	Reserved	Reserved	RO	-
[40:39]	pmu_sn_home_sel	HN-F PMU SN/Home select 2'b00: All requests selected 2'b01: SN bound requests selected 2'b10: Home bound requests selected	RW	2'h0
[38:37]	pmu_hbt_lbt_sel	HN-F PMU HBT/LBT select 2'b00: All requests selected 2'b01: HBT requests selected 2'b10: LBT requests selected	RW	2'h0
[36:35]	pmu_class_occup_id	HN-F PMU Class select **2'b00** Class 0 selected **2'b01** Class 1 selected **2'b10** Class 2 selected **2'b11** Class 3 selected	RW	2'h0

Bits	Name	Description	Type	Reset
[34:32]	pmu_occup1_id	HN-F PMU occupancy 1 select **3'b000** All occupancy selected **3'b001** Read requests **3'b010** Write requests **3'b011** Atomic operation requests **3'b100** Stash requests **3'b101** RxSnp requests **3'b110** LBT requests **3'b111** HBT requests	RW	3'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	HN-F PMU Event 3 select; see pmu_event0_id for encodings	RW	6'h00
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	HN-F PMU Event 2 select; see pmu_event0_id for encodings	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	HN-F PMU Event 1 select; see pmu_event0_id for encodings	RW	6'h00
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	HN-F PMU Event 0 select **6'h00** No event **6'h01** PMU_HN_CACHE_MISS_EVENT; counts total cache misses in first lookup result (high priority). Filtering is programmed in pmu_hbt_lbt_sel **6'h02** PMU_HN_SLCSF_CACHE_ACCESS_EVENT; counts number of cache accesses in first access (high priority) **6'h03** PMU_HN_CACHE_FILL_EVENT; counts total allocations in HN SLC (all cache line allocations to SLC) **6'h04** PMU_HN_POCQ_RETRY_EVENT; counts number of retried requests **6'h05** PMU_HN_POCQ_REQS_RECVD_EVENT; counts number of requests received by HN **6'h06** PMU_HN_SF_HIT_EVENT; counts number of SF hits **6'h07** PMU_HN_SF_EVICTIONS_EVENT; counts number of SF eviction cache invalidations initiated **6'h08** PMU_HN_DIR_SNOOPS_SENT_EVENT; counts number of directed snoops sent (not including SF back invalidation) **6'h09** PMU_HN_BRD_SNOOPS_SENTEVENT; counts number of multicast snoops send (not including SF back invalidation) **6'h0A** PMU_HN_SLC_EVICTION_EVENT; counts number of SLC evictions (dirty only) **6'h0B** PMU_HN_SLC_FILL_INVALID_WAY_EVENT; counts number of SLC fills to an invalid way **6'h0C** PMU_HN_MC_RETRIES_EVENT; counts number of retried transactions by the MC **6'h0D** PMU_HN_MC_REQS_EVENT; counts number of requests sent to MC **6'h0E** PMU_HN_QOS_HH_RETRY_EVENT; counts number of times a HighHigh priority request is protocol retried at the HN-F **6'h0F** PMU_HN_POCQ_OCCUPANCY_EVENT; counts the POCQ occupancy in HN-F; occupancy filtering is programmed in pmu_occup1_id **6'h10** PMU_HN_POCQ_ADDRHAZ_EVENT; counts number of POCQ address hazards upon allocation **6'h11** PMU_HN_POCQ_ATOMICS_ADDRHAZ_EVENT; counts number of POCQ address hazards upon allocation for atomic operations **6'h12** PMU_HN_LD_ST_SWP_ADQ_FULL_EVENT; counts number of times ADQ is full for Ld/St/SWP type atomic operations while POCQ has pending operations **6'h13** PMU_HN_CMP_ADQ_FULL_EVENT; counts number of times ADQ is full for CMP type atomic operations while POCQ has pending operations **6'h14** PMU_HN_TXDAT_STALL_EVENT; counts number of times HN-F has a pending TXDAT flit but no credits to upload **6'h15** PMU_HN_TXRSP_STALL_EVENT; counts number of times HN-F has a pending TXRSP flit but no credits to upload **6'h16** PMU_HN_SEQ_FULL_EVENT; counts number of times requests are replayed in SLC pipe due to SEQ being full **6'h17** PMU_HN_SEQ_HIT_EVENT; counts number of times a request in SLC hit a pending SF eviction in SEQ **6'h18** PMU_HN_SNP_SENT_EVENT; counts number of snoops sent including directed/multicast/SF back invalidation **6'h19** PMU_HN_SFBI_DIR_SNP_SENT_EVENT; counts number of times directed snoops were sent due to SF back invalidation **6'h1a** PMU_HN_SFBI_BRD_SNP_SENT_EVENT; counts number of times multicast snoops were sent due to SF back invalidation **6'h1b** PMU_HN_SNP_SENT_UNTRK_EVENT; counts number of times snooped were sent due to untracked RN-Fs **6'h1c** PMU_HN_INTV_DIRTY_EVENT; counts number of times SF back invalidation resulted in dirty line intervention from the RN	RW	6'h00

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	HN-F PMU Event 0 select **6'h1d** PMU_HN_STASH_SNP_SENT_EVENT; counts number of times stash snoops sent **6'h1e** PMU_HN_STASH_DATA_PULL_EVENT; counts number of times stash snoops resulted in data pull from the RN **6'h1f** PMU_HN_SNP_FWDED_EVENT; counts number of times data forward snoops sent **6'h20** PMU_HN_ATOMIC_FWD_EVENT; counts number of times atomic data was forwarded between POC entries **6'h21** PMU_HN_MPAM_REQ_OVER_HARDLIM_EVENT; counts number of times write req can't allocate in SLC due to being over hardlimit **6'h22** PMU_HN_MPAM_REQ_OVER_SOFTLIM_EVENT; counts number of times write req is above soft limit **6'h23** PMU_HN_SNP_SENT_CLUSTER_EVENT; counts number of snoops sent to clusters excluding indivual snoops within a cluster **6'h24** PMU_HN_SF_IMPRECISE_EVICT_EVENT; counts number of times an evict op was dropped due to SF clustering **6'h25** PMU_HN_SF_EVICT_SHARED_LINE_EVENT; counts number of times a shared line was evicted from SF **6'h26** PMU_HN_POCQ_CLASS_OCCUPANCY_EVENT; counts the POCQ occupancy for a given class in HN-F; Class occupancy filtering is programmed in pmu_class_occup_id **6'h27** PMU_HN_POCQ_CLASS_RETRY_EVENT; counts number of retried requests for a given class; Class filtering is programmed in pmu_class_occup_id **6'h28** PMU_HN_CLASS_MC_REQS_EVENT; counts number of requests sent to MC for a given class; Class filtering is programmed in pmu_class_occup_id **6'h29** PMU_HN_CLASS_PCRDGNT_BELOW_CONDMIN_EVENT; counts number of protocol credit grants for a given class when it's above dedicated and below conditional min; Class filtering is programmed in pmu_class_occup_id **6'h2A** PMU_HN_NUM_SN_CBUSY_THROTTLE_EVENT; counts number of times request to SN was throttled due to cbusy; Event filtering is programmed in pmu_cbusy_snthrottle_sel **6'h2B** PMU_HN_NUM_SN_CBUSY_THROTTLE_MIN_EVENT; counts number of times request to SN was throttled to the minimum due to cbusy; Event filtering is programmed in pmu_cbusy_snthrottle_sel **6'h2C** PMU_HN_SF_PRECISE_TO_IMPRECISE_EVENT; counts when number sharers exceeds how many RN's could be precisely tracked in SF **6'h2D** PMU_HN_SNP_INTV_CLN_EVENT; counts the number of times clean data intervened for a snoop request **6'h2E** PMU_HN_NC_EXCL_EVENT; counts the number of times non-cacheable exclusive request arrived at HNF **6'h2F** PMU_HN_EXCL_MON_OVFL_EVENT; counts the number of times exclusive monitor overflowed	RW	6'h00

4.3.10.111 cmn_hns_pmu_mpam_sel

Specifies details of MPAM event to be counted

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2008`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-392: cmn_hns_pmu_mpam_sel

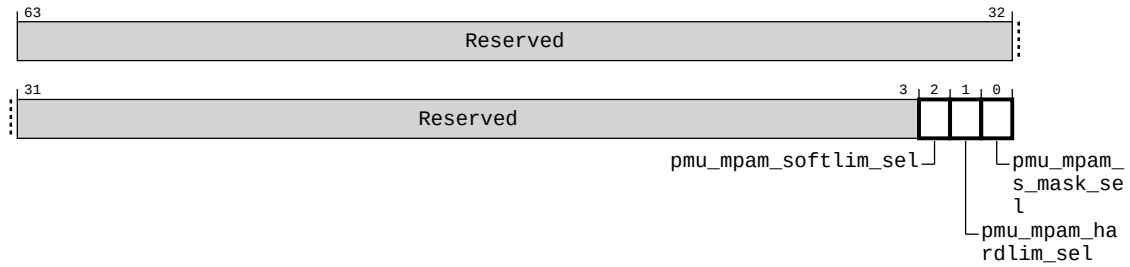


Table 4-408: cmn_hns_pmu_mpam_sel attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	pmu_mpam_softlim_sel	When set, HN-F PMU MPAM Softlimit count is filtered for specific PARTIDs 1'b0: PMU Softlimit count is total for all PARDIDs. 1'b1: PMU Softlimit count is only for PARDIDs indicated in fliter register	RW	1'b0
[1]	pmu_mpam_hardlim_sel	When set, HN-F PMU MPAM Hardlimit count is filtered for specific PARTIDs 1'b0: PMU Hardlimit count is total for all PARDIDs. 1'b1: PMU Hardlimit count is only for PARDIDs indicated in fliter register	RW	1'b0
[0]	pmu_mpam_s_mask_sel	When set, PARTID Mask is used for Secure MPAM PARTID 1'b0: PMU MPAM mask is for Non-secure MPAMID. 1'b1: PMU MPAM mask is for Secure MPAMID.	RW	1'b0

4.3.10.112 cmn_hns_pmu_mpam_pardid_mask0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Functions as mask for PARTID[#{64*(index+1)-1}:#{64*index}] filter for MPM PMU events

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h2010 + #{8*index}
index(0) : 16'h0
```

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-393: cmn_hns_pmu_mpam_pardid_mask0-7

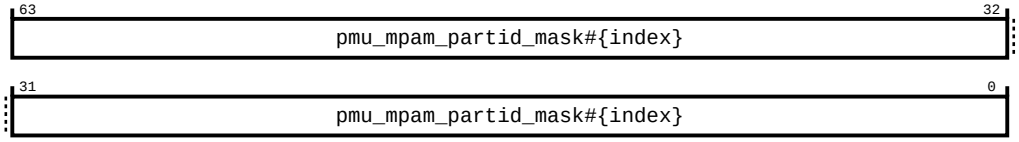


Table 4-409: cmn_hns_pmu_mpam_pardid_mask0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	pmu_mpam_partid_mask#{index}	MPAM PMU hardlimit and softlimit mask for PARTID [#{64*(index +1)-1}:#{64*index}] **1'b0** PARTID specified is not counted in PMU count. **1'b1** PARTID specified is counted in PMU count. This mask is used only when cmn_hns_pmu_mpam_sel is set for PARTID based counting. Note: This mask is used only when cmn_hns_pmu_mpam_sel is set for PARTID based counting.	RW	64'b0

4.3.10.113 cmn_hns_rn_cluster0-63_physid_reg0

There are 64 iterations of this register. The index ranges from 0 to 63. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3C00 + #{index}*32`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-394: cmn_hns_rn_cluster0-63_physid_reg0

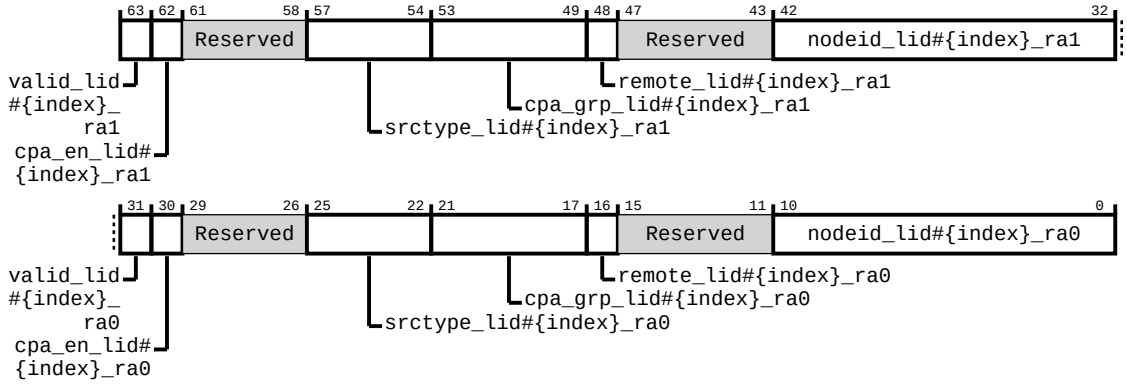


Table 4-410: cmn_hns_rn_cluster0-63_physid_reg0 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra1	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra1	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[61:58]	Reserved	Reserved	RO	-
[57:54]	srctype_lid#{index}_ra1	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[53:49]	cpa_grp_lid#{index}_ra1	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra1	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra1	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra0	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra0	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[29:26]	Reserved	Reserved	RO	-
[25:22]	srctype_lid#{index}_ra0	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[21:17]	cpa_grp_lid#{index}_ra0	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra0	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra0	Specifies the node ID	RW	11'h0

4.3.10.114 cmn_hns_rn_cluster64-127_physid_reg0

There are 64 iterations of this register. The index ranges from 64 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3C00 + #{index}*32`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-395: cmn_hns_rn_cluster64-127_physid_reg0

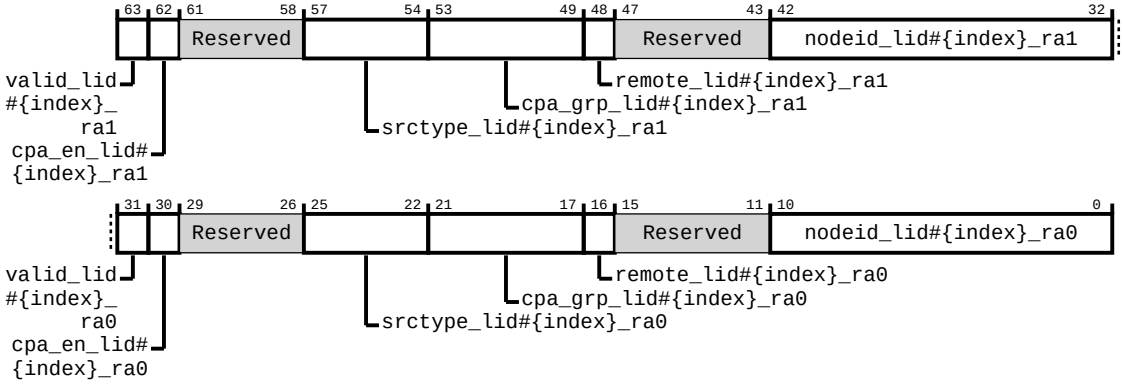


Table 4-411: cmn_hns_rn_cluster64-127_physid_reg0 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra1	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra1	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[61:58]	Reserved	Reserved	RO	-
[57:54]	srctype_lid#{index}_ra1	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[53:49]	cpa_grp_lid#{index}_ra1	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra1	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra1	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra0	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra0	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[29:26]	Reserved	Reserved	RO	-
[25:22]	srctype_lid#{index}_ra0	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[21:17]	cpa_grp_lid#{index}_ra0	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra0	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra0	Specifies the node ID	RW	11'h0

4.3.10.115 cmn_hns_rn_cluster0-127_physid_reg1

There are 128 iterations of this register. The index ranges from 0 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3C08 + #{index}*32`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-396: cmn_hns_rn_cluster0-127_physid_reg1

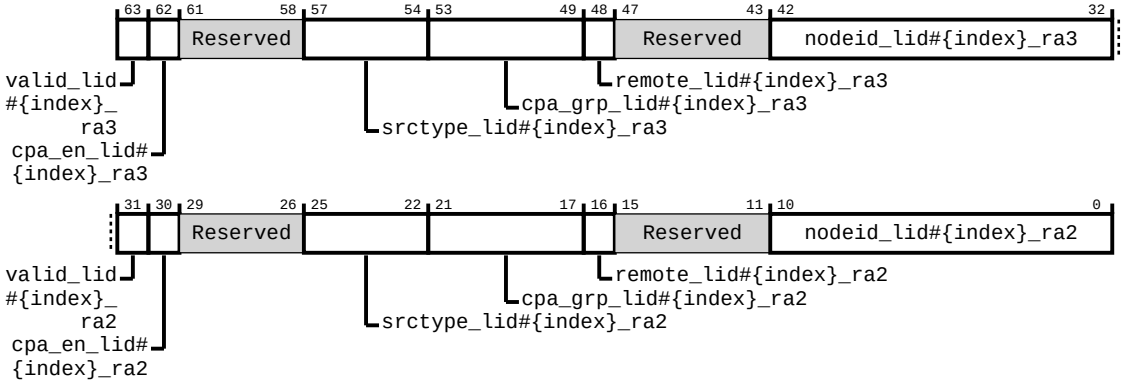


Table 4-412: cmn_hns_rn_cluster0-127_physid_reg1 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra3	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra3	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[61:58]	Reserved	Reserved	RO	-
[57:54]	srctype_lid#{index}_ra3	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[53:49]	cpa_grp_lid#{index}_ra3	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra3	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra3	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra2	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra2	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[29:26]	Reserved	Reserved	RO	-
[25:22]	srctype_lid#{index}_ra2	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[21:17]	cpa_grp_lid#{index}_ra2	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra2	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra2	Specifies the node ID	RW	11'h0

4.3.10.116 cmn_hns_rn_cluster0-127_physid_reg2

There are 128 iterations of this register. The index ranges from 0 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3C10 + #{index}*32`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-397: cmn_hns_rn_cluster0-127_physid_reg2

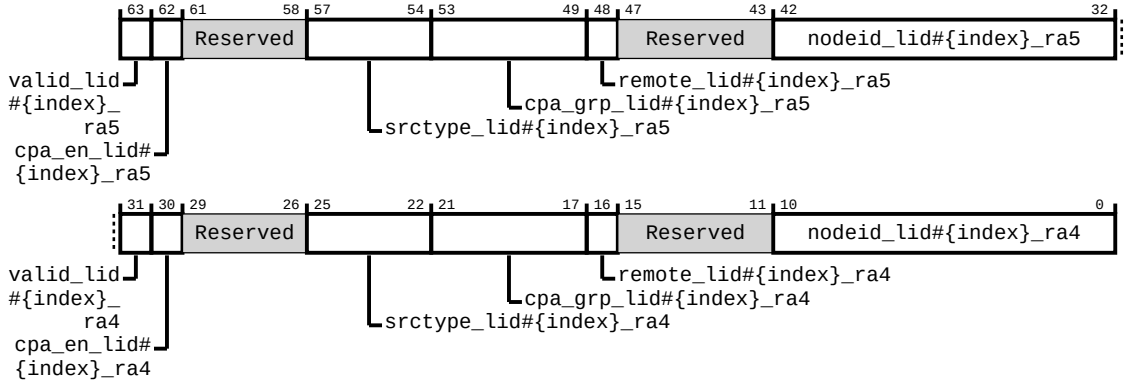


Table 4-413: cmn_hns_rn_cluster0-127_physid_reg2 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra5	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra5	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[61:58]	Reserved	Reserved	RO	-
[57:54]	srctype_lid#{index}_ra5	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[53:49]	cpa_grp_lid#{index}_ra5	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra5	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra5	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra4	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra4	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[29:26]	Reserved	Reserved	RO	-
[25:22]	srctype_lid#{index}_ra4	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[21:17]	cpa_grp_lid#{index}_ra4	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra4	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra4	Specifies the node ID	RW	11'h0

4.3.10.117 cmn_hns_rn_cluster0-127_physid_reg3

There are 128 iterations of this register. The index ranges from 0 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3C18 + #{index}*32`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-398: cmn_hns_rn_cluster0-127_physid_reg3

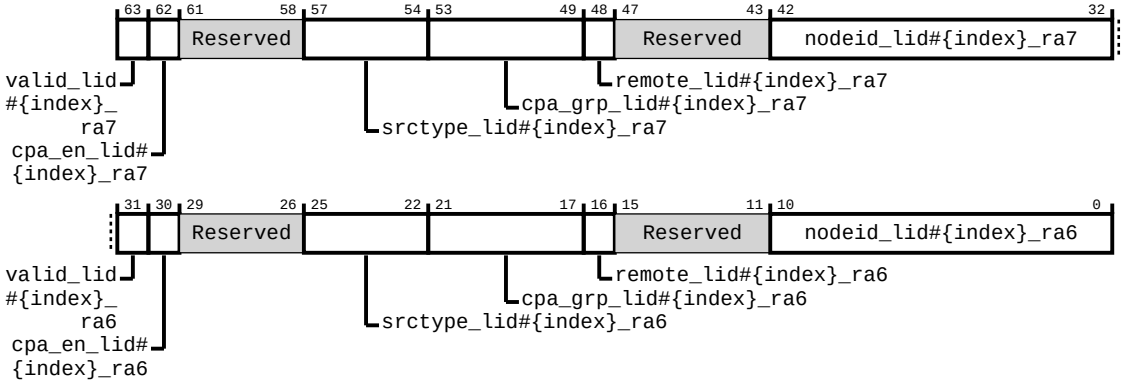


Table 4-414: cmn_hns_rn_cluster0-127_physid_reg3 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra7	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra7	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[61:58]	Reserved	Reserved	RO	-
[57:54]	srctype_lid#{index}_ra7	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[53:49]	cpa_grp_lid#{index}_ra7	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra7	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra7	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra6	Specifies whether the RN is valid **1'b0** RN ID is not valid **1'b1** RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra6	Specifies whether the CCIX port aggregation is enabled **1'b0** CPA not enabled **1'b1** CPA enabled	RW	1'h0
[29:26]	Reserved	Reserved	RO	-
[25:22]	srctype_lid#{index}_ra6	Specifies the CHI source type of the RN **4'b1010** 256 bit CHI-B RN-F **4'b1011** 256 bit CHI-C RN-F **4'b1100** 256 bit CHI-D RN-F **4'b1101** 256 bit CHI-E RN-F **Others** Reserved	RW	4'h0000
[21:17]	cpa_grp_lid#{index}_ra6	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra6	Specifies whether the RN is remote or local **1'b0** Local RN **1'b1** Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra6	Specifies the node ID	RW	11'h0

4.3.10.118 cmn_hns_sam_nonhash_cfg1_memregion2-63

There are 62 iterations of this register. The index ranges from 2 to 63. Configures non-hashed memory region #{index} in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5000 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-399: cmn_hns_sam_nonhash_cfg1_memregion2-63

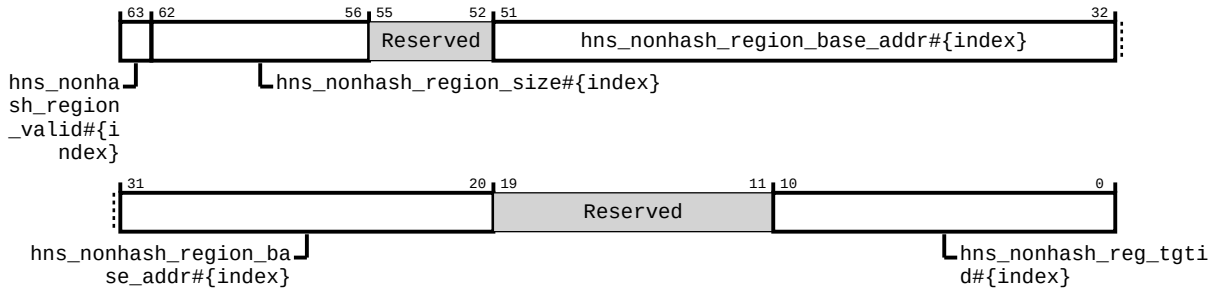


Table 4-415: cmn_hns_sam_nonhash_cfg1_memregion2-63 attributes

Bits	Name	Description	Type	Reset
[63]	hns_nonhash_region_valid#{index}	valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0
[62:56]	hns_nonhash_region_size#{index}	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b0
[55:52]	Reserved	Reserved	RO	-
[51:20]	hns_nonhash_region_base_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:11]	Reserved	Reserved	RO	-
[10:0]	hns_nonhash_reg_tgtid#{index}	SN TgtID for the non-hashed region	RW	11'h0

4.3.10.119 cmn_hns_sam_nonhash_cfg2_memregion2-63

There are 62 iterations of this register. The index ranges from 2 to 63. Configures non-hashed memory region #{index} in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5200 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-400: cmn_hns_sam_nonhash_cfg2_memregion2-63

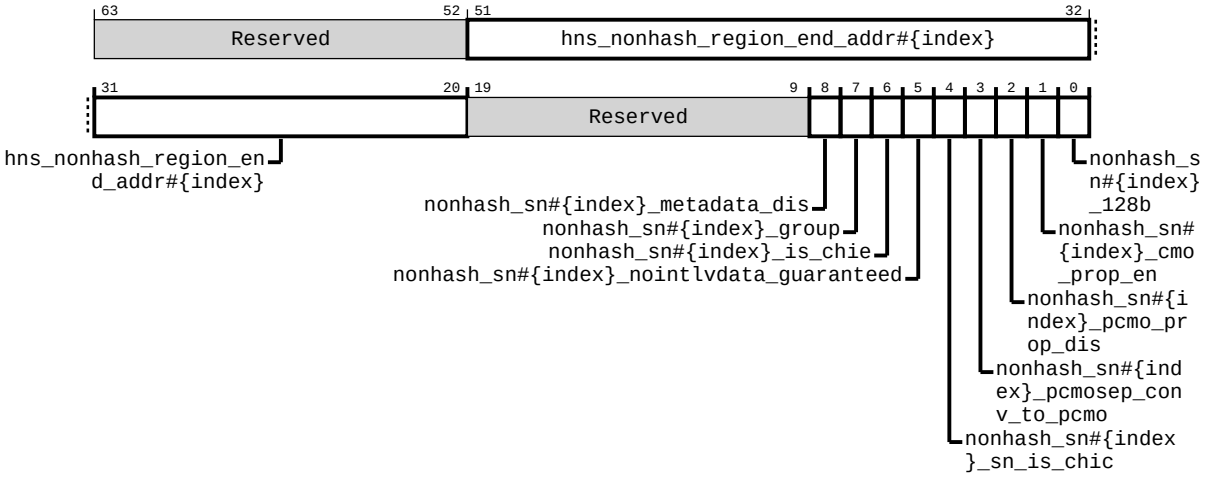


Table 4-416: cmn_hns_sam_nonhash_cfg2_memregion2-63 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	hns_nonhash_region_end_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:9]	Reserved	Reserved	RO	-
[8]	nonhash_sn#{index}_metadata_dis	HNS implements metadata termination flow for nonhash SN #{index} when set	RW	1'b0
[7]	nonhash_sn#{index}_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[6]	nonhash_sn#{index}_is_chie	nonhash SN #{index} supports CHI-E	RW	1'b0
[5]	nonhash_sn#{index}_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[4]	nonhash_sn#{index}_sn_is_chic	Indicates that nonhash sn is a CHI-C SN when set	RW	1'b0
[3]	nonhash_sn#{index}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for nonhash SN #{index} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	nonhash_sn#{index}_pcmo_prop_dis	Disables PCMO propagation for nonhash SN #{index} when set	RW	1'b0
[1]	nonhash_sn#{index}_cmo_prop_en	Enables CMO propagation for nonhash SN #{index} when set	RW	1'b0
[0]	nonhash_sn#{index}_128b	Data width of nonhash SN #{index} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0

4.3.10.120 cmn_hns_sam_htg_cfg1_memregion0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures HTG memory region #{index} in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5400 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-401: cmn_hns_sam_htg_cfg1_memregion0-15

Table 4-417: cmn_hns_sam_htg_cfg1_memregion0-15 attributes

Bits	Name	Description	Type	Reset
[63]	htg_region_valid#{index}	valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0
[62:56]	htg_region_size#{index}	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b00000
[55:52]	Reserved	Reserved	RO	-
[51:20]	htg_region_base_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

4.3.10.121 cmn_hns_sam_htg_cfg2_memregion0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures htg memory region #{index} in HN-F SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5480 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-402: cmn_hns_sam_htg_cfg2_memregion0-15

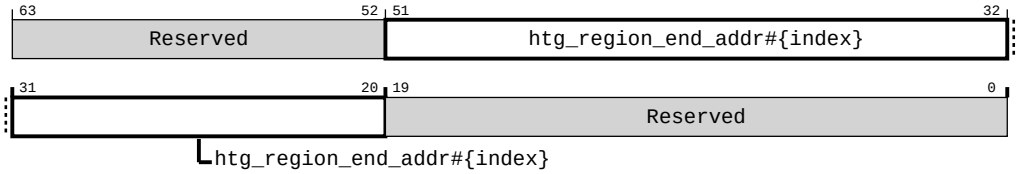


Table 4-418: cmn_hns_sam_htg_cfg2_memregion0-15 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	htg_region_end_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

4.3.10.122 cmn_hns_sam_htg_cfg3_memregion0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the HTG memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5500 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-403: cmn_hns_sam_htg_cfg3_memregion0-15

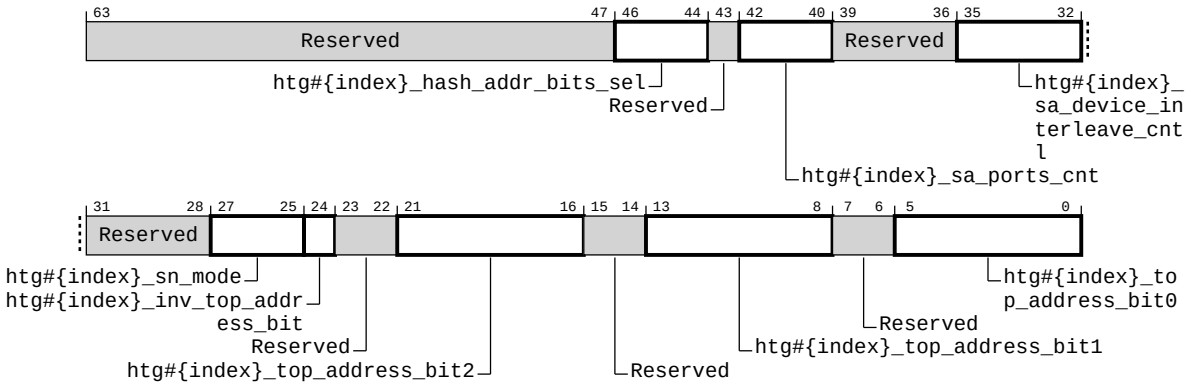


Table 4-419: cmn_hns_sam_htg_cfg3_memregion0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:44]	htg#{index}_hash_addr_bits_sel	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[43]	Reserved	Reserved	RO	-
[42:40]	htg#{index}_sa_ports_cnt	Specifies the number of CXSA/CXLSA device aggregated **1'b0** Local RN **1'b1** Remote RN	RW	3'b0
[39:36]	Reserved	Reserved	RO	-
[35:32]	htg#{index}_sa_device_interleave_cntl	This field controls the interleave size across all aggregated CXSA/CXLSA Devices **4'h0** 64B Interleaved **4'h1** 128B Interleaved **4'h2** 256B Interleaved **4'h3** 512B Interleaved **4'hF** 2MB Interleaved	RW	4'b0

Bits	Name	Description	Type	Reset
[31:28]	Reserved	Reserved	RO	-
[27:25]	htg#{index}_sn_mode	SN selection mode **3'b000** Reserved **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **3'b111** CXSA/CXLSA aggregated SA selection function	RW	3'b0
[24]	htg#{index}_inv_top_address_bit	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	htg#{index}_top_address_bit2	Top address bit 2	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg#{index}_top_address_bit1	Top address bit 1	RW	6'h00
[7:6]	Reserved	Reserved	RO	-
[5:0]	htg#{index}_top_address_bit0	Top address bit 0	RW	6'h00

4.3.10.123 cmn_hns_sam_htg_sn_nodeid_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures SN node IDs for HTGs in the HNSAM . Controls target SN node IDs #{index*4 + 0} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5600 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-404: cmn_hns_sam_htg_sn_nodeid_reg0-15

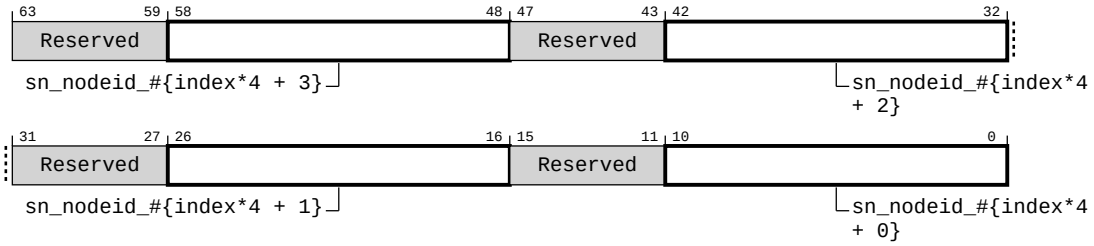


Table 4-420: cmn_hns_sam_htg_sn_nodeid_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	sn_nodeid_#{index*4 + 3}	Hashed target SN node ID #{index*4 + 3}	RW	11'b00000000000
[47:43]	Reserved	Reserved	RO	-
[42:32]	sn_nodeid_#{index*4 + 2}	Hashed target SN node ID #{index*4 + 2}	RW	11'b00000000000
[31:27]	Reserved	Reserved	RO	-
[26:16]	sn_nodeid_#{index*4 + 1}	Hashed target SN node ID #{index*4 + 1}	RW	11'b00000000000
[15:11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_#{index*4 + 0}	Hashed target SN node ID #{index*4 + 0}	RW	11'b00000000000

4.3.10.124 cmn_hns_sam_htg_sn_attr0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures SN node attributes HTGs in the HNSAM . Controls SN attributes #{index*4 + 0} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5680 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-405: cmn_hns_sam_htg_sn_attr0-15

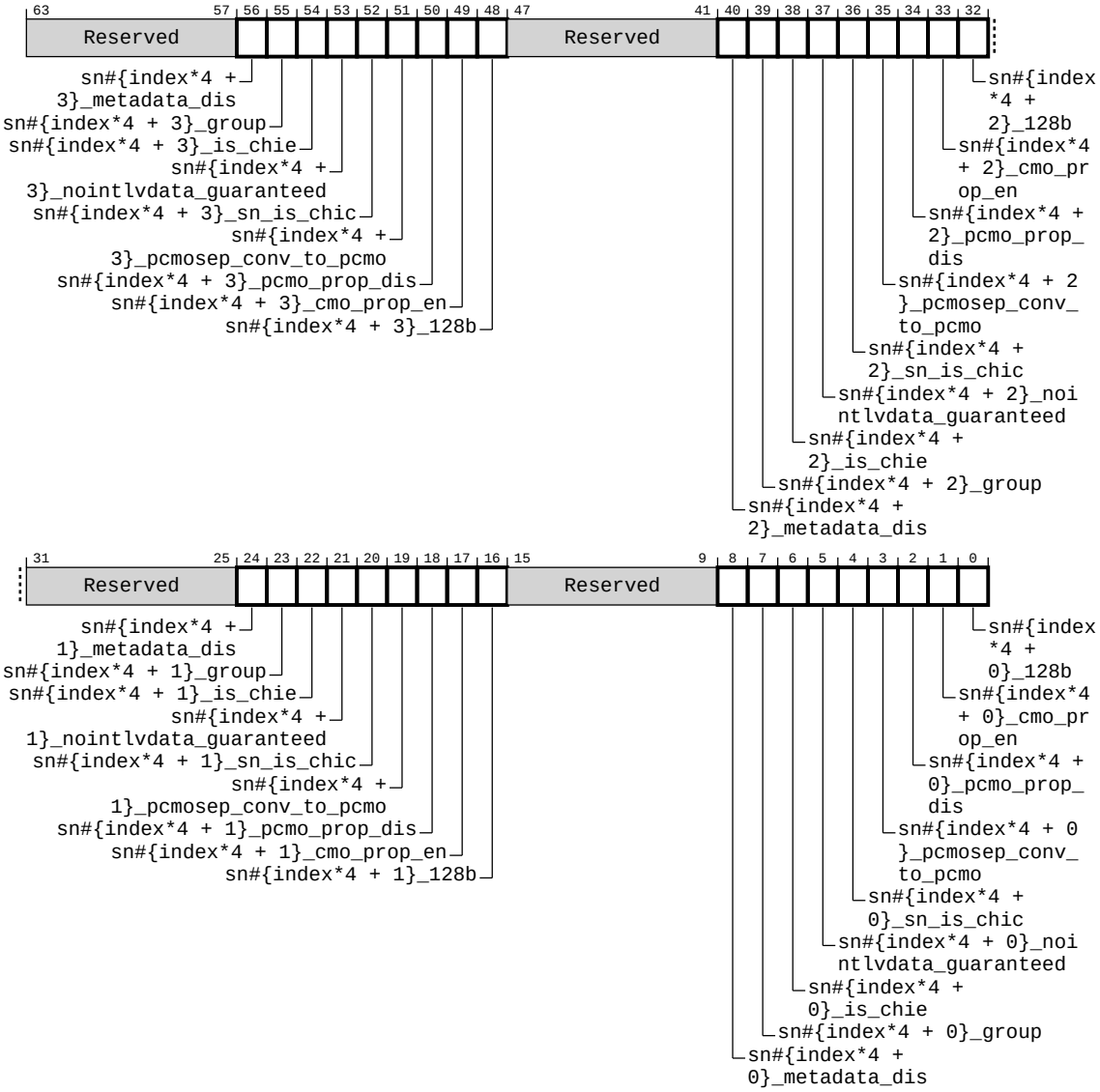


Table 4-421: cmn_hns_sam_htg_sn_attr0-15 attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56]	sn#{index*4 + 3}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 3} when set	RW	1'b0
[55]	sn#{index*4 + 3}_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[54]	sn#{index*4 + 3}_is_chie	SN #{index*4 + 3} supports CHI-E	RW	1'b0

Bits	Name	Description	Type	Reset
[53]	sn#{index*4 + 3}_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[52]	sn#{index*4 + 3}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[51]	sn#{index*4 + 3}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 3} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[50]	sn#{index*4 + 3}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 3} when set	RW	1'b0
[49]	sn#{index*4 + 3}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 3} when set	RW	1'b0
[48]	sn#{index*4 + 3}_128b	Data width of SN #{index*4 + 3} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[47:41]	Reserved	Reserved	RO	-
[40]	sn#{index*4 + 2}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 2} when set	RW	1'b0
[39]	sn#{index*4 + 2}_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[38]	sn#{index*4 + 2}_is_chie	SN #{index*4 + 2} supports CHI-E	RW	1'b0
[37]	sn#{index*4 + 2}_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[36]	sn#{index*4 + 2}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[35]	sn#{index*4 + 2}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 2} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[34]	sn#{index*4 + 2}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 2} when set	RW	1'b0
[33]	sn#{index*4 + 2}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 2} when set	RW	1'b0
[32]	sn#{index*4 + 2}_128b	Data width of SN #{index*4 + 2} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[31:25]	Reserved	Reserved	RO	-
[24]	sn#{index*4 + 1}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 1} when set	RW	1'b0
[23]	sn#{index*4 + 1}_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[22]	sn#{index*4 + 1}_is_chie	SN #{index*4 + 1} supports CHI-E	RW	1'b0
[21]	sn#{index*4 + 1}_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[20]	sn#{index*4 + 1}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[19]	sn#{index*4 + 1}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 1} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0

Bits	Name	Description	Type	Reset
[18]	sn#{index*4 + 1}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 1} when set	RW	1'b0
[17]	sn#{index*4 + 1}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 1} when set	RW	1'b0
[16]	sn#{index*4 + 1}_128b	Data width of SN #{index*4 + 1} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[15:9]	Reserved	Reserved	RO	-
[8]	sn#{index*4 + 0}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 0} when set	RW	1'b0
[7]	sn#{index*4 + 0}_group	Specifies the SN-F grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[6]	sn#{index*4 + 0}_is_chie	SN #{index*4 + 0} supports CHI-E	RW	1'b0
[5]	sn#{index*4 + 0}_nointlvdata_guaranteed	SN guaratees the return data will not be interleaved	RW	1'b0
[4]	sn#{index*4 + 0}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[3]	sn#{index*4 + 0}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 0} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	sn#{index*4 + 0}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 0} when set	RW	1'b0
[1]	sn#{index*4 + 0}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 0} when set	RW	1'b0
[0]	sn#{index*4 + 0}_128b	Data width of SN #{index*4 + 0} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0

4.3.10.125 cmn_hns_sam_ccg_sa_nodeid_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures CCG SA node IDs for HTGs in the HNSAM

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h5700 + #{index}*8
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-406: cmn_hns_sam_ccg_sa_nodeid_reg0-3

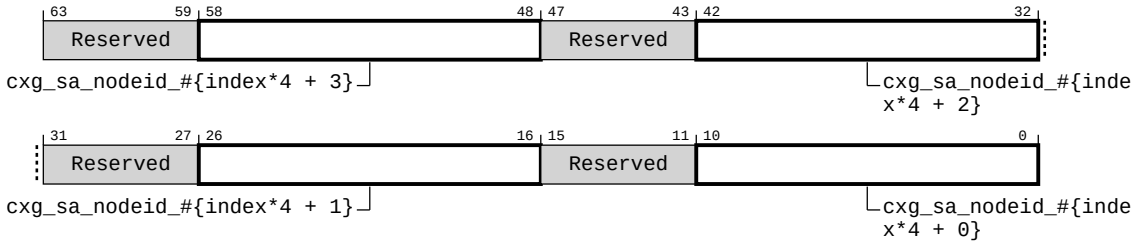


Table 4-422: cmn_hns_sam_ccg_sa_nodeid_reg0-3 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	cxg_sa_nodeid_#{index*4 + 3}	Hashed target CCG SA node ID #{index*4 + 3}	RW	11'b00000000000
[47:43]	Reserved	Reserved	RO	-
[42:32]	cxg_sa_nodeid_#{index*4 + 2}	Hashed target CCG SA node ID #{index*4 + 2}	RW	11'b00000000000
[31:27]	Reserved	Reserved	RO	-
[26:16]	cxg_sa_nodeid_#{index*4 + 1}	Hashed target CCG SA node ID #{index*4 + 1}	RW	11'b00000000000
[15:11]	Reserved	Reserved	RO	-
[10:0]	cxg_sa_nodeid_#{index*4 + 0}	Hashed target CCG SA node ID #{index*4 + 0}	RW	11'b00000000000

4.3.10.126 cmn_hns_sam_ccg_sa_attr0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures CCG SA node attributes.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5740 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-407: cmn_hns_sam_ccg_sa_attr0-3

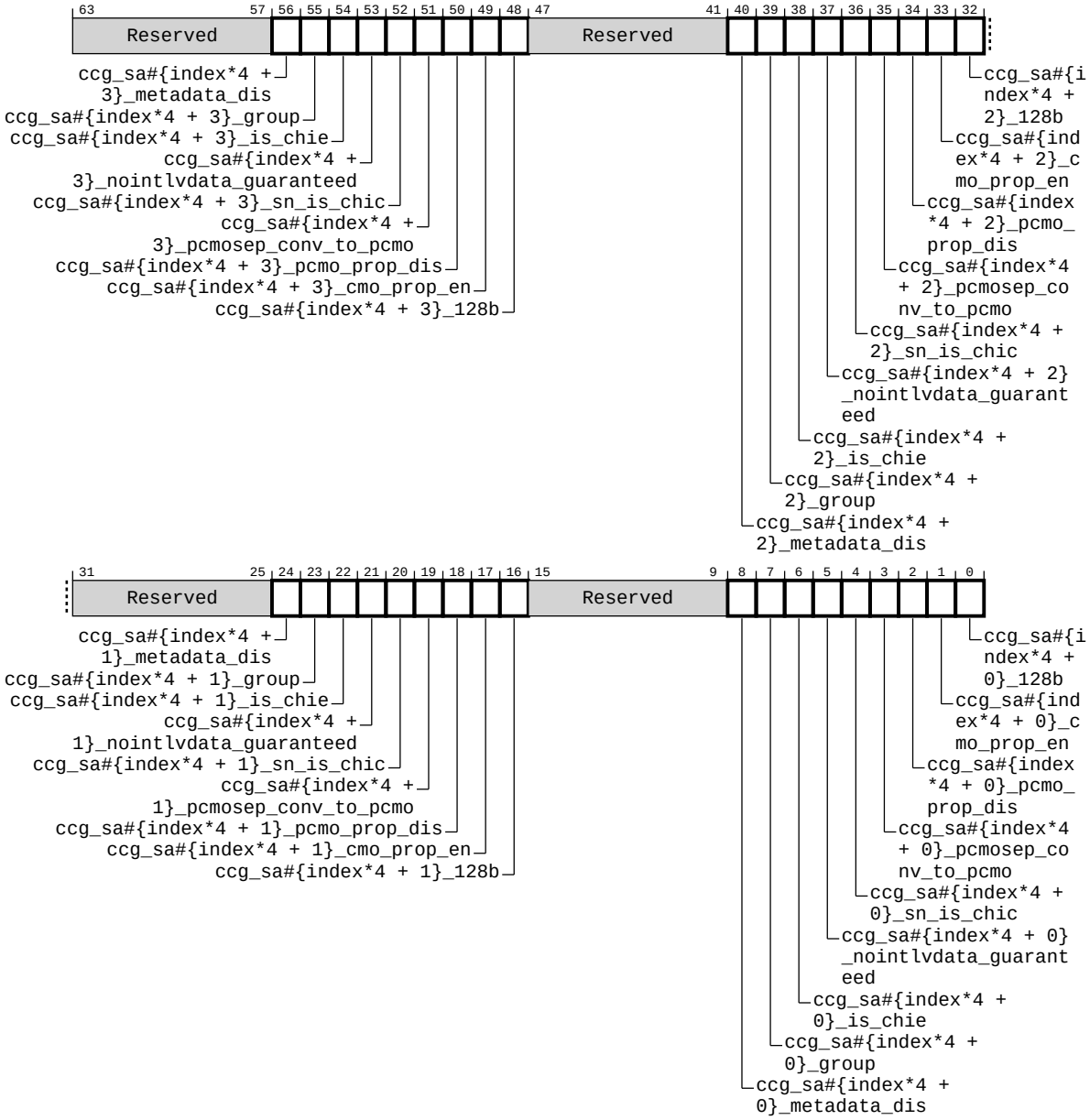


Table 4-423: cmn_hns_sam_ccg_sa_attr0-3 attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56]	ccg_sa#{index*4 + 3}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 3} when set	RW	1'b0
[55]	ccg_sa#{index*4 + 3}_group	Specifies the CCG_SA grouping **1'b0** Group A **1'b1** Group B	RW	1'b0

Bits	Name	Description	Type	Reset
[54]	ccg_sa#{index*4 + 3}_is_chie	CCG_SA #{index*4 + 3} supports CHI-E	RW	1'b0
[53]	ccg_sa#{index*4 + 3}_nointlvdata_guaranteed	CCG_SA guaratees the return data will not be interleaved	RW	1'b0
[52]	ccg_sa#{index*4 + 3}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[51]	ccg_sa#{index*4 + 3}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 3} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[50]	ccg_sa#{index*4 + 3}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 3} when set	RW	1'b0
[49]	ccg_sa#{index*4 + 3}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 3} when set	RW	1'b0
[48]	ccg_sa#{index*4 + 3}_128b	Data width of CCG_SA #{index*4 + 3} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[47:41]	Reserved	Reserved	RO	-
[40]	ccg_sa#{index*4 + 2}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 2} when set	RW	1'b0
[39]	ccg_sa#{index*4 + 2}_group	Specifies the CCG_SA grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[38]	ccg_sa#{index*4 + 2}_is_chie	CCG_SA #{index*4 + 2} supports CHI-E	RW	1'b0
[37]	ccg_sa#{index*4 + 2}_nointlvdata_guaranteed	CCG_SA guaratees the return data will not be interleaved	RW	1'b0
[36]	ccg_sa#{index*4 + 2}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[35]	ccg_sa#{index*4 + 2}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 2} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[34]	ccg_sa#{index*4 + 2}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 2} when set	RW	1'b0
[33]	ccg_sa#{index*4 + 2}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 2} when set	RW	1'b0
[32]	ccg_sa#{index*4 + 2}_128b	Data width of CCG_SA #{index*4 + 2} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[31:25]	Reserved	Reserved	RO	-
[24]	ccg_sa#{index*4 + 1}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 1} when set	RW	1'b0
[23]	ccg_sa#{index*4 + 1}_group	Specifies the CCG_SA grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[22]	ccg_sa#{index*4 + 1}_is_chie	CCG_SA #{index*4 + 1} supports CHI-E	RW	1'b0

Bits	Name	Description	Type	Reset
[21]	ccg_sa#{index*4 + 1}_nointlvdata_guaranteed	CCG_SA guaratees the return data will not be interleaved	RW	1'b0
[20]	ccg_sa#{index*4 + 1}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[19]	ccg_sa#{index*4 + 1}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 1} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[18]	ccg_sa#{index*4 + 1}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 1} when set	RW	1'b0
[17]	ccg_sa#{index*4 + 1}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 1} when set	RW	1'b0
[16]	ccg_sa#{index*4 + 1}_128b	Data width of CCG_SA #{index*4 + 1} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0
[15:9]	Reserved	Reserved	RO	-
[8]	ccg_sa#{index*4 + 0}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 0} when set	RW	1'b0
[7]	ccg_sa#{index*4 + 0}_group	Specifies the CCG_SA grouping **1'b0** Group A **1'b1** Group B	RW	1'b0
[6]	ccg_sa#{index*4 + 0}_is_chie	CCG_SA #{index*4 + 0} supports CHI-E	RW	1'b0
[5]	ccg_sa#{index*4 + 0}_nointlvdata_guaranteed	CCG_SA guaratees the return data will not be interleaved	RW	1'b0
[4]	ccg_sa#{index*4 + 0}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[3]	ccg_sa#{index*4 + 0}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 0} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	ccg_sa#{index*4 + 0}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 0} when set	RW	1'b0
[1]	ccg_sa#{index*4 + 0}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 0} when set	RW	1'b0
[0]	ccg_sa#{index*4 + 0}_128b	Data width of CCG_SA #{index*4 + 0} **1'b1** 128 bits **1'b0** 256 bits	RW	1'b0

4.3.10.127 hns_generic_regs0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configuration register for the custom logic

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5780 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-408: hns_generic_regs0-7

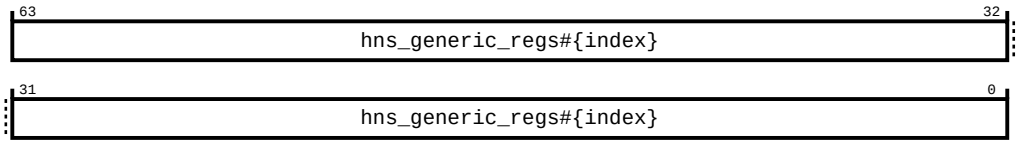


Table 4-424: hns_generic_regs0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	hns_generic_regs#{index}	Configuration register for the custom logic	RW	64'h0

4.3.10.128 cmn_hns_pa2setaddr_slc

Functions as the control register of PA to SetAddr and vice versa conversion for HNS-SLC

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h5900

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-409: cmn_hns_pa2setaddr_slc

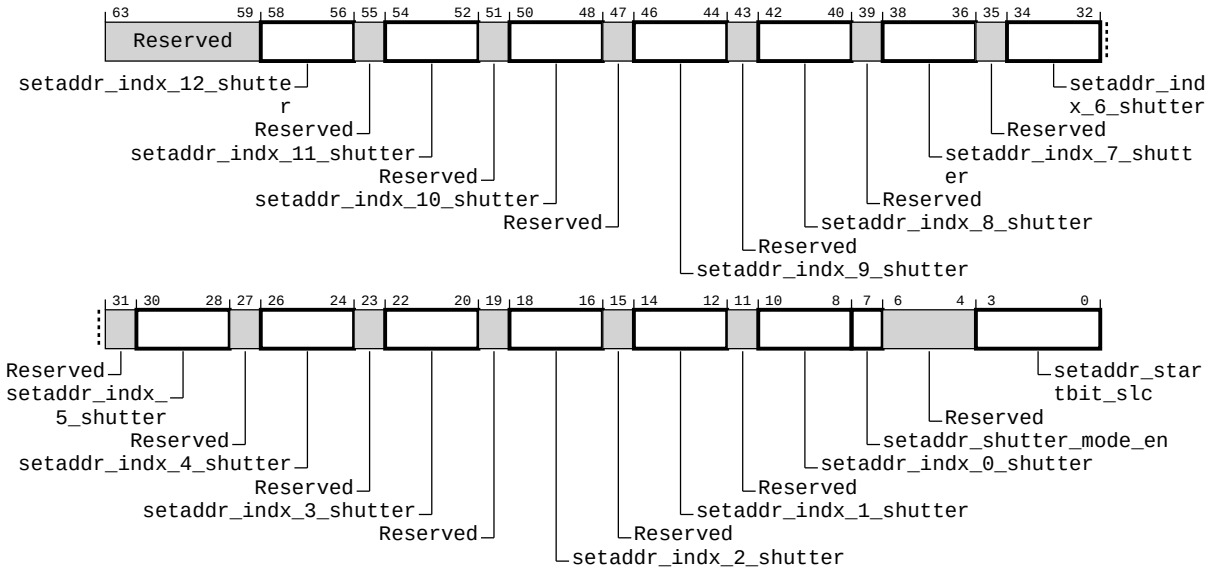


Table 4-425: cmn_hns_pa2setaddr_slc attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[58:56]	setaddr_indx_12_shutter	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_indx_11_shutter	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_indx_10_shutter	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_indx_9_shutter	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_indx_8_shutter	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:36]	setaddr_indx_7_shutter	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[35]	Reserved	Reserved	RO	-
[34:32]	setaddr_indx_6_shutter	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_indx_5_shutter	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_indx_4_shutter	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_indx_3_shutter	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[19]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[18:16]	setaddr_indx_2_shutter	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_indx_1_shutter	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[11]	Reserved	Reserved	RO	-
[10:8]	setaddr_indx_0_shutter	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[7]	setaddr_shutter_mode_en	Enables address shuttering mode for SLC as programmed by setaddr_indx_X_shutter registers	RW	1'b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_slc	SLC: SetAddr starting bit for SLC TODO add a description here abt contiguous bits **4'b0110** Setaddr starts from PA[6] **4'b0111** Setaddr starts from PA[7] **4'b1000** Setaddr starts from PA[8] **4'b1001** Setaddr starts from PA[9] **4'b1010** Setaddr starts from PA[10] **4'b1011** Setaddr starts from PA[11] **4'b1100** Setaddr starts from PA[12]	RW	4'b0110

4.3.10.129 cmn_hns_pa2setaddr_sf

Functions as the control register of PA to Set/TagAddr and vice versa conversion for HNS-SF

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5908`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-410: cmn_hns_pa2setaddr_sf

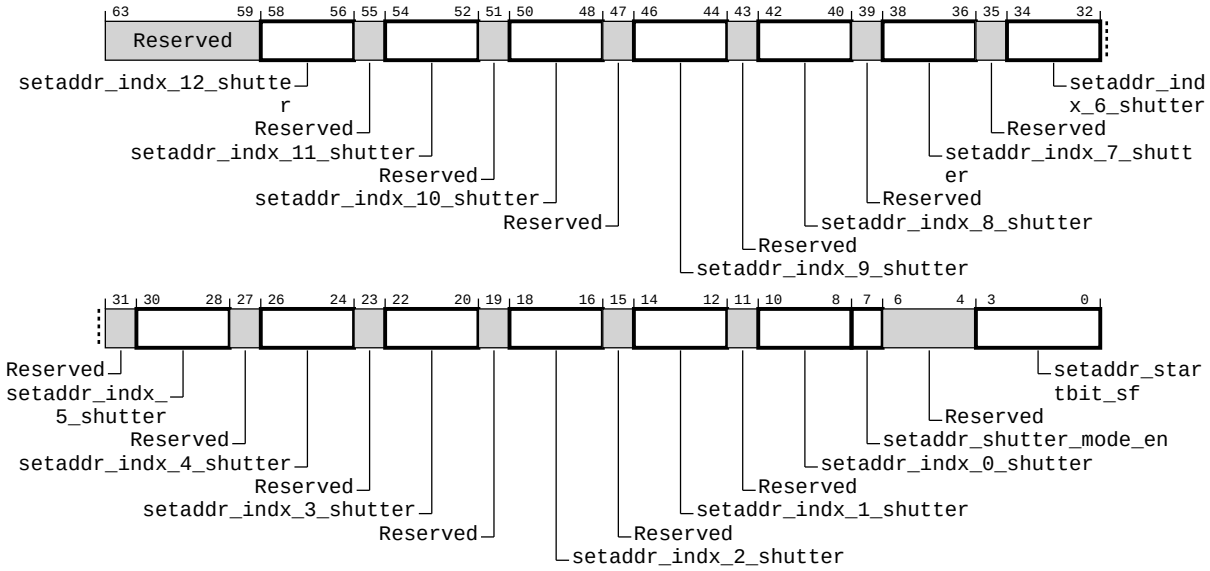


Table 4-426: cmn_hns_pa2setaddr_sf attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[58:56]	setaddr_indx_12_shutter	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_indx_11_shutter	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_indx_10_shutter	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_indx_9_shutter	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_indx_8_shutter	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:36]	setaddr_indx_7_shutter	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[35]	Reserved	Reserved	RO	-
[34:32]	setaddr_indx_6_shutter	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_indx_5_shutter	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_indx_4_shutter	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_indx_3_shutter	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[19]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[18:16]	setaddr_indx_2_shutter	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_indx_1_shutter	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[11]	Reserved	Reserved	RO	-
[10:8]	setaddr_indx_0_shutter	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[7]	setaddr_shutter_mode_en	Enables address shuttering mode for SF as programmed by setaddr_indx_X_shutter registers	RW	1'b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_sf	SF: SetAddr starting bit for SF **4'b0110** Setaddr starts from PA[6] **4'b0111** Setaddr starts from PA[7] **4'b1000** Setaddr starts from PA[8] **4'b1001** Setaddr starts from PA[9] **4'b1010** Setaddr starts from PA[10] **4'b1011** Setaddr starts from PA[11] **4'b1100** Setaddr starts from PA[12]	RW	4'b0110

4.3.10.130 cmn_hns_pa2setaddr_flex_slc

Functions as the SLC control register of PA to Set/TagAddr and vice versa conversion for HNS (flexible)

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5910`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-411: cmn_hns_pa2setaddr_flex_slc

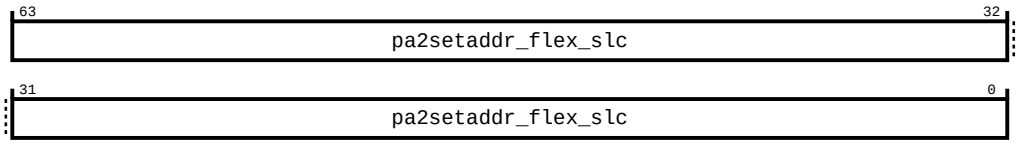


Table 4-427: cmn_hns_pa2setaddr_flex_slc attributes

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_slc	FLEXIBLE: PA to SET/TAG ADDR and vice versa conversion config field for SLC	RW	64'b0

4.3.10.131 cmn_hns_pa2setaddr_flex_sf

Functions as the SF control register of PA to Set/TagAddr and vice versa conversion for HNS (flexible)

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h5918`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-412: cmn_hns_pa2setaddr_flex_sf

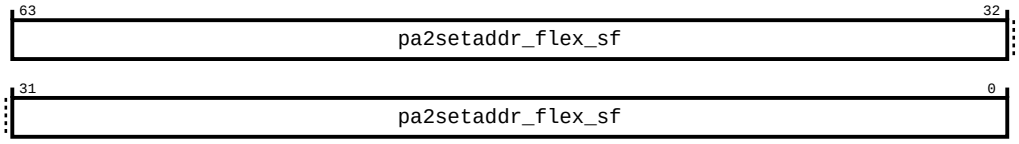


Table 4-428: cmn_hns_pa2setaddr_flex_sf attributes

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_sf	FLEXIBLE: PA to SET/TAG ADDR conversion and vice versa config field for SF	RW	64'b0

4.3.10.132 lcn_hashed_tgt_grp_cfg1_region0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`index(0-31) : 16'h7000 + #{8 * index}`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-413: lcn_hashed_tgt_grp_cfg1_region0-31

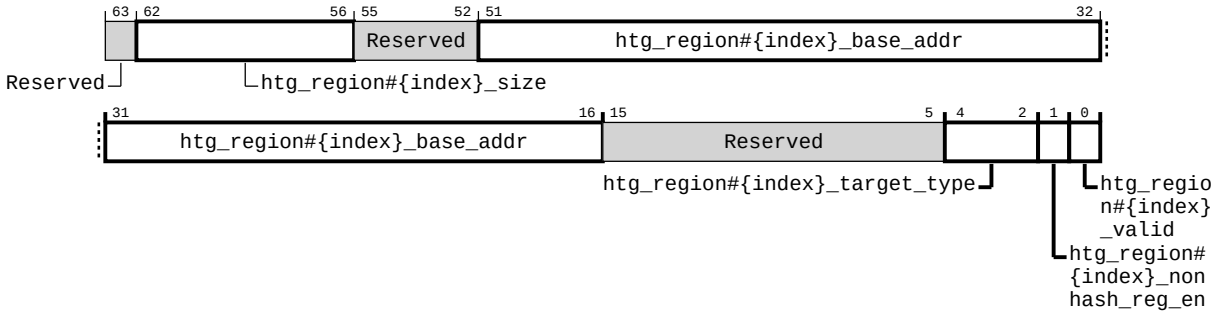


Table 4-429: lcn_hashed_tgt_grp_cfg1_region0-31 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_region#{index}_size	Memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b0000000
[55:52]	Reserved	Reserved	RO	-
[51:16]	htg_region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:2]	htg_region#{index}_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	htg_region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	1'b0
[0]	htg_region#{index}_valid	Memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.10.133 lcn_hashed_tgt_grp_cfg2_region0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h7100 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-414: lcn_hashed_tgt_grp_cfg2_region0-31

Table 4-430: lcn_hashed_tgt_grp_cfg2_region0-31 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_end_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:0]	Reserved	Reserved	RO	-

4.3.10.134 lcn_hashed_target_grp_secondary_cfg1_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures secondary hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h7200 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-415: lcn_hashed_target_grp_secondary_cfg1_reg0-31

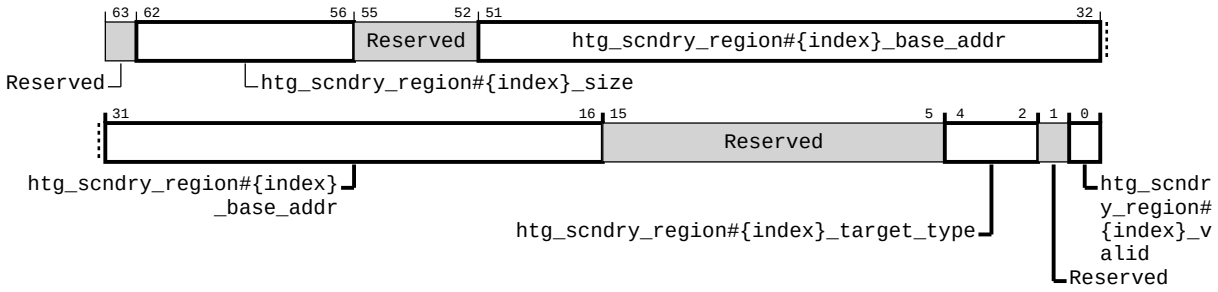


Table 4-431: lcn_hashed_target_grp_secondary_cfg1_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_scndry_region#{index}_size	Secondary memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:2]	htg_scndry_region#{index}_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	Reserved	Reserved	RO	-
[0]	htg_scndry_region#{index}_valid	Secondary memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.10.135 lcn_hashed_target_grp_secondary_cfg2_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h7300 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-416: lcn_hashed_target_grp_secondary_cfg2_reg0-31

Table 4-432: lcn_hashed_target_grp_secondary_cfg2_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_end_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'b000000000000000000000000000000
[15:0]	Reserved	Reserved	RO	-

4.3.10.136 lcn_hashed_target_grp_hash_cntl_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures HTG hash type

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h7400 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-417: lcn_hashed_target_grp_hash_cntl_reg0-31

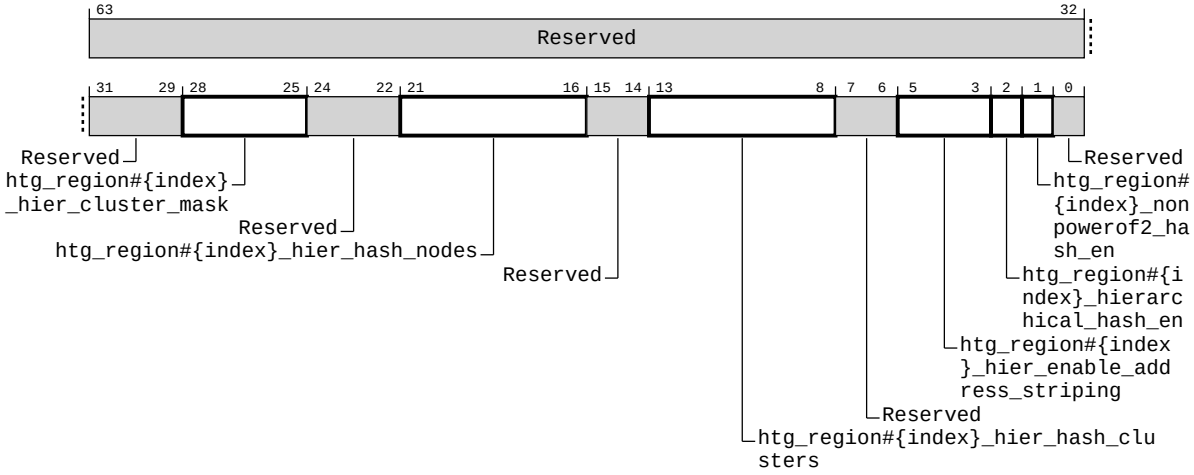


Table 4-433: lcn_hashed_target_grp_hash_cntl_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:25]	htg_region#{index}_hier_cluster_mask	Hierarchical hashing: Enable cluster masking to achieve different interleave granularity across clusters. **4'b0000** 64 byte interleave granularity across clusters **4'b0001** 128 byte interleave granularity across clusters **4'b0010** 256 byte interleave granularity across clusters **4'b0011** 512 byte interleave granularity across clusters **4'b0100** 1024 byte interleave granularity across clusters **4'b0101** 2048 byte interleave granularity across clusters **4'b0110** 4096 byte interleave granularity across clusters **4'b0111** 8192 byte interleave granularity across clusters **Others** Reserved	RW	4'b0
[24:22]	Reserved	Reserved	RO	-
[21:16]	htg_region#{index}_hier_hash_nodes	Hierarchical hashing mode, define number of nodes in each cluster	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg_region#{index}_hier_hash_clusters	Hierarchical hashing mode, define number of clusters groups	RW	6'h0
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:3]	htg_region#{index}_hier_enable_address_striping	Hierarchical hashing: configure number of address bits needs to shuttered (removed) at second hierarchy hash (LSB bit is based on cluster mask). **3'b000** No address shuttering **3'b001** One addr bit shuttered (2 clusters) **3'b010** Two addr bit shuttered (4 clusters) **3'b011** Three addr bit shuttered (8 clusters) **3'b100** Four addr bit shuttered (16 clusters) **3'b101** Five addr bit shuttered (32 clusters) **Others** Reserved	RW	3'b0
[2]	htg_region#{index}_hierarchical_hash_en	Hierarchical Hashing mode enable configure bit	RW	1'b0
[1]	htg_region#{index}_nonpowerof2_hash_en	Non power of two Hashing mode enable cconfigure bit	RW	1'b0
[0]	Reserved	Reserved	RO	-

4.3.10.137 lcn_hashed_target_group_hn_count_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Indicates number of HN-F/HN-P's in hashed target groups #{index*8} to #{index*8 + 7}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-3) : 16'h7500 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-418: lcn_hashed_target_group_hn_count_reg0-3

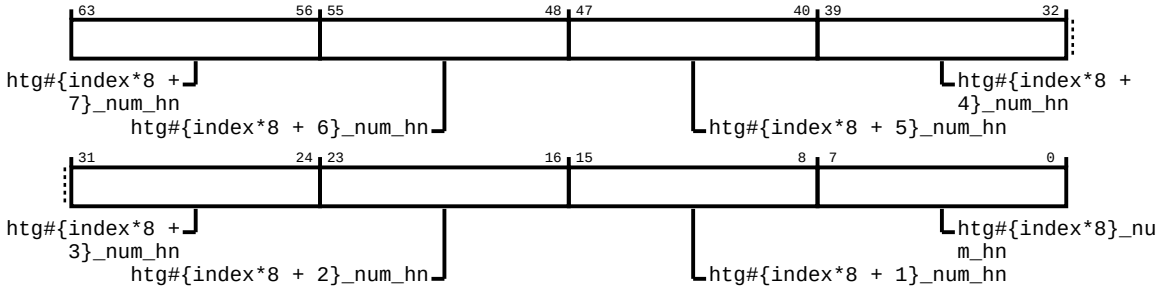


Table 4-434: lcn_hashed_target_group_hn_count_reg0-3 attributes

Bits	Name	Description	Type	Reset
[63:56]	htg#{index*8 + 7}_num_hn	HN count for hashed target group 7	RW	8'h00
[55:48]	htg#{index*8 + 6}_num_hn	HN count for hashed target group 6	RW	8'h00
[47:40]	htg#{index*8 + 5}_num_hn	HN count for hashed target group 5	RW	8'h00
[39:32]	htg#{index*8 + 4}_num_hn	HN count for hashed target group 4	RW	8'h00
[31:24]	htg#{index*8 + 3}_num_hn	HN count for hashed target group 3	RW	8'h00
[23:16]	htg#{index*8 + 2}_num_hn	HN count for hashed target group 2	RW	8'h00
[15:8]	htg#{index*8 + 1}_num_hn	HN count for hashed target group 1	RW	8'h00
[7:0]	htg#{index*8}_num_hn	HN count for hashed target group 0	RW	8'h00

4.3.10.138 lcn_hashed_target_grp_cal_mode_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures the HN CAL mode support for all hashed target groups.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'h7520 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-419: lcn_hashed_target_grp_cal_mode_reg0-7

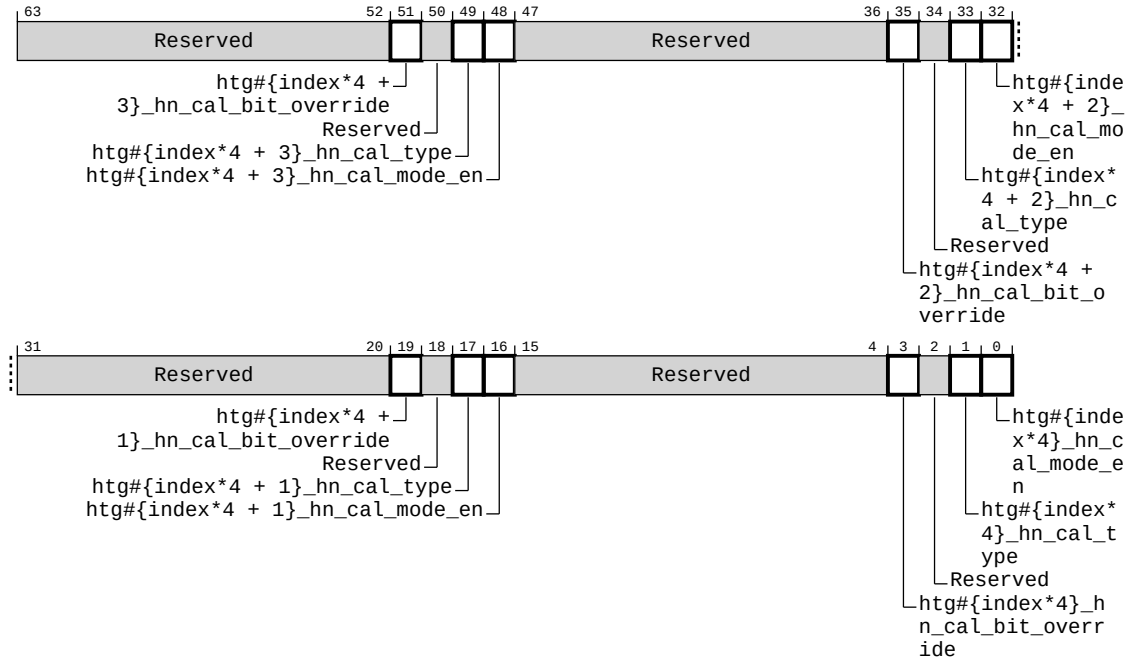


Table 4-435: lcn_hashed_target_grp_cal_mode_reg0-7 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	htg#{index*4 + 3}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 3} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[50]	Reserved	Reserved	RO	-
[49]	htg#{index*4 + 3}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 3} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[48]	htg#{index*4 + 3}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 3}	RW	1'b0
[47:36]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[35]	htg#{index*4 + 2}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 2} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[34]	Reserved	Reserved	RO	-
[33]	htg#{index*4 + 2}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 2} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[32]	htg#{index*4 + 2}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 2}	RW	1'b0
[31:20]	Reserved	Reserved	RO	-
[19]	htg#{index*4 + 1}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 1} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[18]	Reserved	Reserved	RO	-
[17]	htg#{index*4 + 1}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 1} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[16]	htg#{index*4 + 1}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 1}	RW	1'b0
[15:4]	Reserved	Reserved	RO	-
[3]	htg#{index*4}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[2]	Reserved	Reserved	RO	-
[1]	htg#{index*4}_hn_cal_type	Enables type of HN CAL for HTG #{index*4} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[0]	htg#{index*4}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4}	RW	1'b0

4.3.10.139 lcn_hashed_target_grp_hnf_cpa_en_reg0-1

There are 2 iterations of this register. The index ranges from 0 to 1. Configures CCIX port aggregation mode for hashed HNF node IDs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-1) : 16'h7560 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-420: lcn_hashed_target_grp_hnf_cpa_en_reg0-1

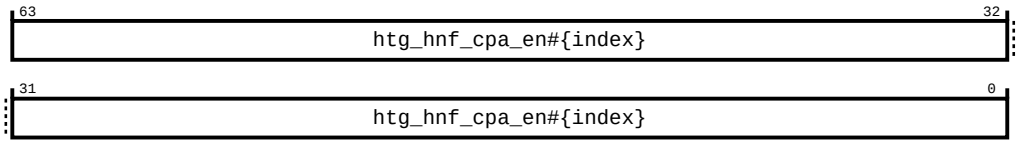


Table 4-436: lcn_hashed_target_grp_hnf_cpa_en_reg0-1 attributes

Bits	Name	Description	Type	Reset
[63:0]	htg_hnf_cpa_en#{index}	Enable CPA for each hashed HNF node ID	RW	64'h0000000000000000

4.3.10.140 lcn_hashed_target_grp_cpag_perhnf_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures CPAG ID for each hashed HNF node IDs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-15) : 16'h7580 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-421: lcn_hashed_target_grp_cpag_perhnf_reg0-15

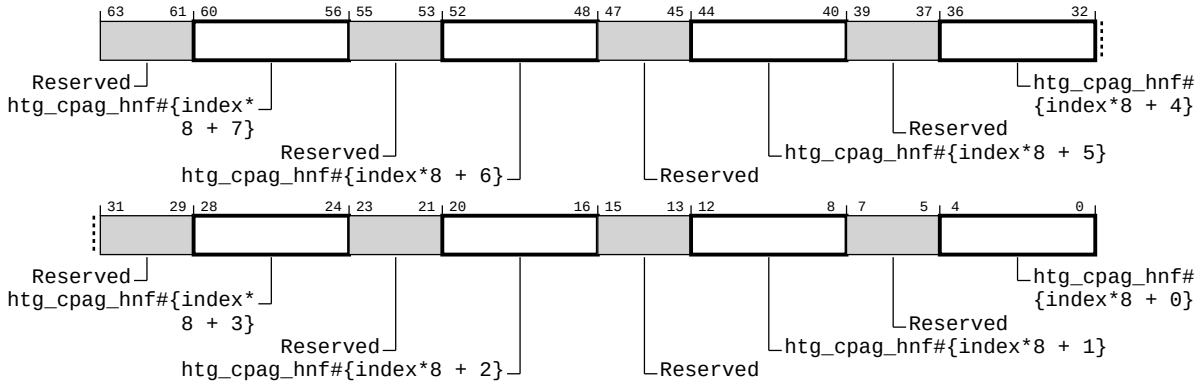


Table 4-437: lcn_hashed_target_grp_cpag_perhnf_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:56]	htg_cpag_hnf#{index*8 + 7}	CPAG associated to the HNF#{index*8 + 7}	RW	5'b0
[55:53]	Reserved	Reserved	RO	-
[52:48]	htg_cpag_hnf#{index*8 + 6}	CPAG associated to the HNF#{index*8 + 6}	RW	5'b0
[47:45]	Reserved	Reserved	RO	-
[44:40]	htg_cpag_hnf#{index*8 + 5}	CPAG associated to the HNF#{index*8 + 5}	RW	5'b0
[39:37]	Reserved	Reserved	RO	-
[36:32]	htg_cpag_hnf#{index*8 + 4}	CPAG associated to the HNF#{index*8 + 4}	RW	5'b0
[31:29]	Reserved	Reserved	RO	-
[28:24]	htg_cpag_hnf#{index*8 + 3}	CPAG associated to the HNF#{index*8 + 3}	RW	5'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	htg_cpag_hnf#{index*8 + 2}	CPAG associated to the HNF#{index*8 + 2}	RW	5'b0
[15:13]	Reserved	Reserved	RO	-
[12:8]	htg_cpag_hnf#{index*8 + 1}	CPAG associated to the HNF#{index*8 + 1}	RW	5'b0
[7:5]	Reserved	Reserved	RO	-
[4:0]	htg_cpag_hnf#{index*8 + 0}	CPAG associated to the HNF#{index*8 + 0}	RW	5'b0

4.3.10.141 lcn_hashed_target_grp_compact_cpag_ctrl0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the CPAG control for HTG#{index} valid only when POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h7700 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-422: lcn_hashed_target_grp_compact_cpag_ctrl0-31

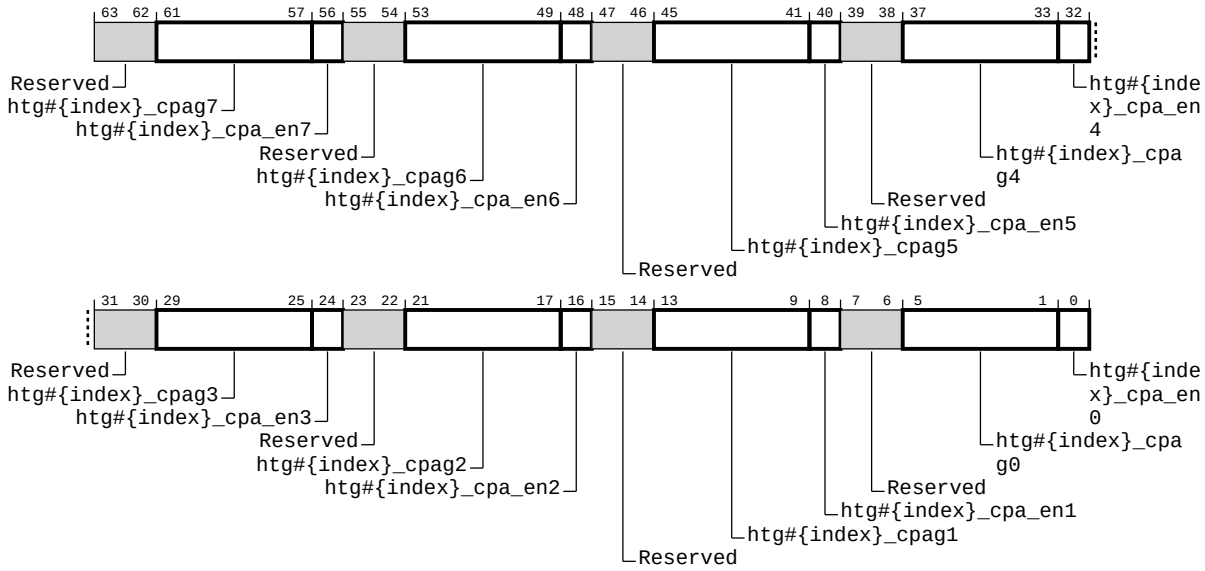


Table 4-438: lcn_hashed_target_grp_compact_cpag_ctrl0-31 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:57]	htg#{index}_cpag7	cpag id for index7	RW	5'b0
[56]	htg#{index}_cpa_en7	cpa enable for index7	RW	1'b0
[55:54]	Reserved	Reserved	RO	-
[53:49]	htg#{index}_cpag6	cpag id for index6	RW	5'b0
[48]	htg#{index}_cpa_en6	cpa enable for index6	RW	1'b0
[47:46]	Reserved	Reserved	RO	-
[45:41]	htg#{index}_cpag5	cpag id for index5	RW	5'b0
[40]	htg#{index}_cpa_en5	cpa enable for index5	RW	1'b0
[39:38]	Reserved	Reserved	RO	-
[37:33]	htg#{index}_cpag4	cpag id for index4	RW	5'b0
[32]	htg#{index}_cpa_en4	cpa enable for index4	RW	1'b0
[31:30]	Reserved	Reserved	RO	-
[29:25]	htg#{index}_cpag3	cpag id for index0	RW	5'b0
[24]	htg#{index}_cpa_en3	cpa enable for index3	RW	1'b0
[23:22]	Reserved	Reserved	RO	-
[21:17]	htg#{index}_cpag2	cpag id for index2	RW	5'b0
[16]	htg#{index}_cpa_en2	cpa enable for index2	RW	1'b0
[15:14]	Reserved	Reserved	RO	-
[13:9]	htg#{index}_cpag1	cpag id for index1	RW	5'b0
[8]	htg#{index}_cpa_en1	cpa enable for index1	RW	1'b0
[7:6]	Reserved	Reserved	RO	-
[5:1]	htg#{index}_cpag0	cpag id for index0	RW	5'b0
[0]	htg#{index}_cpa_en0	cpa enable for index0	RW	1'b0

4.3.10.142 lcn_hashed_target_grp_compact_hash_ctrl0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the HNF hash selecetion and CPAG hash selection control information for HTG#{index} valid only when POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h7800 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-423: lcn_hashed_target_grp_compact_hash_ctrl0-31

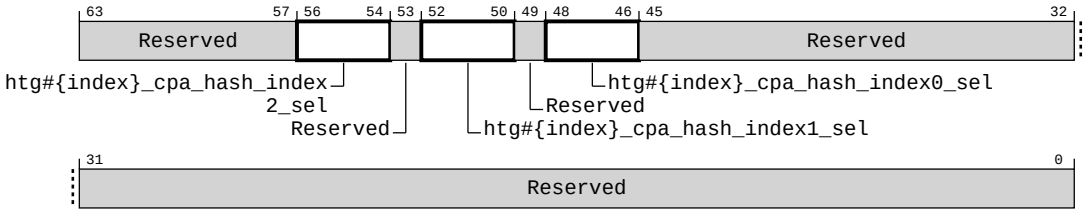


Table 4-439: lcn_hashed_target_grp_compact_hash_ctrl0-31 attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:54]	htg#{index}_cpa_hash_index2_sel	configures the CPAG hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index2. **3'b001** SMP hash index2 + 1. **3'b010** SMP hash index2 + 2. **3'b011** SMP hash index2 + 3. **3'b100** SMP hash index2 + 4. **3'b101** SMP hash index2 + 5 **3'b110** SMP hash index2 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[53]	Reserved	Reserved	RO	-
[52:50]	htg#{index}_cpa_hash_index1_sel	configures the CPAG hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index1. **3'b001** SMP hash index1 + 1. **3'b010** SMP hash index1 + 2. **3'b011** SMP hash index1 + 3. **3'b100** SMP hash index1 + 4. **3'b101** SMP hash index1 + 5 **3'b110** SMP hash index1 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[49]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[48:46]	htg#{index}_cpa_hash_index0_sel	configures the CPAG hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index0. **3'b001** SMP hash index0 + 1. **3'b010** SMP hash index0 + 2. **3'b011** SMP hash index0 + 3. **3'b100** SMP hash index0 + 4. **3'b101** SMP hash index0 + 5 **3'b110** SMP hash index0 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[45:0]	Reserved	Reserved	RO	-

4.3.11 HN-F MPAM_NS register descriptions

This section lists the HN-F MPAM_NS registers.

4.3.11.1 cmn_hns_mpam_ns_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-424: cmn_hns_mpam_ns_node_info

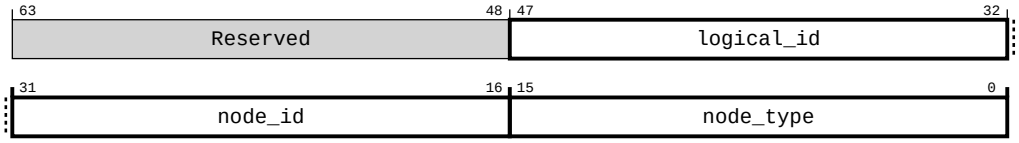


Table 4-440: cmn_hns_mpam_ns_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	$logical_id_description	RO	Configuration dependent
[31:16]	node_id	$node_id_description	RO	Configuration dependent
[15:0]	node_type	$node_type_description	RO	Configuration dependent

4.3.11.2 cmn_hns_mpam_ns_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-425: cmn_hns_mpam_ns_child_info

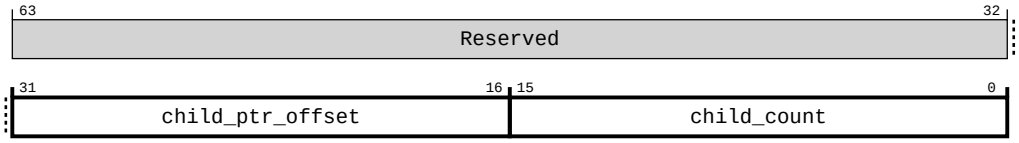


Table 4-441: cmn_hns_mpam_ns_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.11.3 cmn_hns_mpam_idr

MPAM features ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1000`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-426: cmn_hns_mpam_idr

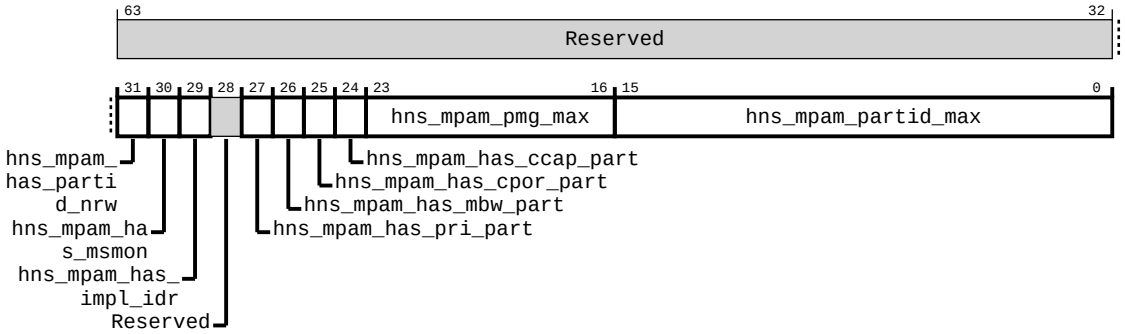


Table 4-442: cmn_hns_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_has_partid_nrw	**0** HN-F does not support MPAM PARTID Narrowing **1** HN-F supports MPAM PARTID Narrowing	RO	Configuration dependent
[30]	hns_mpam_has_msmon	**0** MPAM performance monitoring is not supported **1** MPAM performance monitoring is supported	RO	Configuration dependent
[29]	hns_mpam_has_impl_idr	**0** MPAM implementation specific partitioning features not supported **1** MPAM implementation specific partitioning features supported	RO	Configuration dependent
[28]	Reserved	Reserved	RO	-
[27]	hns_mpam_has_pri_part	**0** MPAM priority partitioning is not supported **1** MPAM priority partitioning is supported	RO	Configuration dependent
[26]	hns_mpam_has_mbw_part	**0** MPAM memory bandwidth partitioning is not supported **1** MPAM memory bandwidth partitioning is supported	RO	Configuration dependent
[25]	hns_mpam_has_cpor_part	**0** MPAM cache portion partitioning is not supported **1** MPAM cache portion partitioning is supported	RO	Configuration dependent
[24]	hns_mpam_has_ccap_part	**0** MPAM cache maximum capacity partitioning is not supported **1** MPAM cache maximum capacity partitioning is supported	RO	Configuration dependent
[23:16]	hns_mpam_pmg_max	Maximum value of Non-secure PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_mpam_partid_max	Maximum value of Non-secure PARTID supported by this HN-F	RO	Configuration dependent

4.3.11.4 cmn_hns_mpam_iidr

MPAM Implementation ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1018`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-427: cmn_hns_mpam_iidr

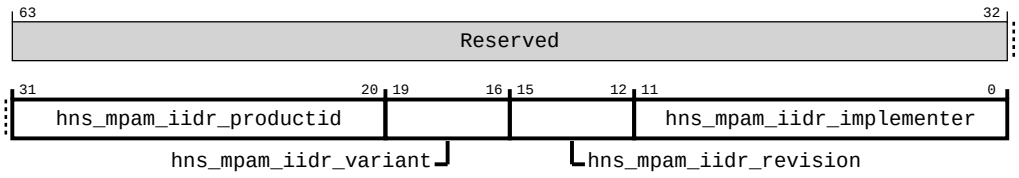


Table 4-443: cmn_hns_mpam_iidr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hns_mpam_iidr_productid	Implementation defined value identifying MPAM memory system component	RO	12'h000
[19:16]	hns_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	4'b0000
[15:12]	hns_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	4'b0000
[11:0]	hns_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	12'h43B

4.3.11.5 cmn_hns_mpam_aidr

MPAM architecture ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1020`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-428: cmn_hns_mpam_aidr

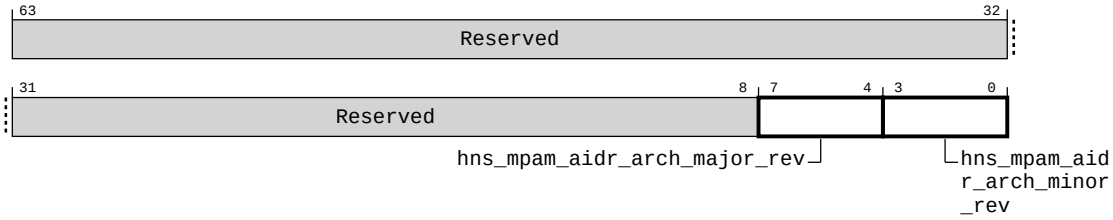


Table 4-444: cmn_hns_mpam_aidr attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hns_mpam_aidr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	4'b0001
[3:0]	hns_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	4'b0000

4.3.11.6 cmn_hns_mpam_impl_idr

MPAM Implementation defined partitioning feature ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1028`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-429: cmn_hns_mpam_impl_idr

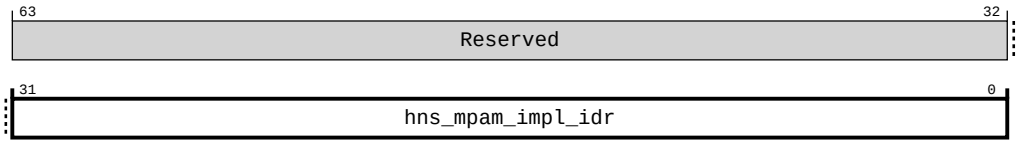


Table 4-445: cmn_hns_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hns_mpam_impl_idr	Implementation defined partitioning features.	RO	32'h00000000

4.3.11.7 cmn_hns_mpam_cpor_idr

MPAM cache portion partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1030`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-430: cmn_hns_mpam_cpor_idr

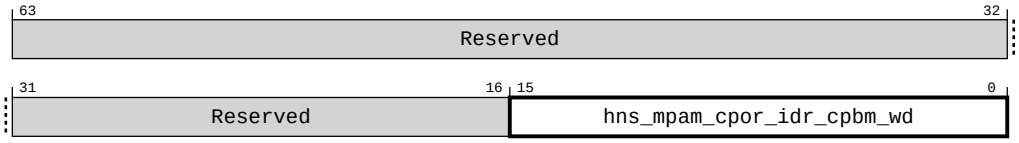


Table 4-446: cmn_hns_mpam_cpor_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bit map of this device.	RO	Configuration dependent

4.3.11.8 cmn_hns_mpam_ccap_idr

MPAM cache capacity partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1038`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-431: cmn_hns_mpam_ccap_idr

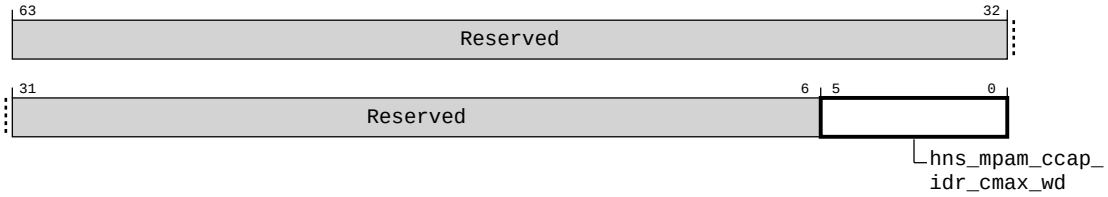


Table 4-447: cmn_hns_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:0]	hns_mpam_ccap_idr_cmax_wd	Number of fractional bits implemented in the cache capacity partitioning.	RO	Configuration dependent

4.3.11.9 cmn_hns_mpam_mbw_idr

MPAM Memory Bandwidth partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1040`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-432: cmn_hns_mpam_mbw_idr

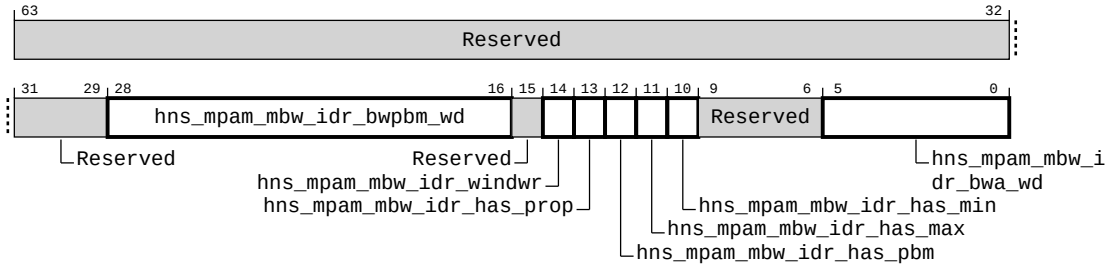


Table 4-448: cmn_hns_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hns_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register.	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hns_mpam_mbw_idr_windwr	**0** The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed **1** The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.	RO	1'h0
[13]	hns_mpam_mbw_idr_has_prop	**0** There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register **1** MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.	RO	1'h0
[12]	hns_mpam_mbw_idr_has_pbm	**0** There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register **1** MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.	RO	1'h0
[11]	hns_mpam_mbw_idr_has_max	**0** There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register **1** MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.	RO	1'h0
[10]	hns_mpam_mbw_idr_has_min	**0** There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register **1** MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.	RO	1'h0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_mpam_mbw_idr_bwa_wd	Number of implemented bits in bandwidth allocation fields: MIN, MAX, and STRIDE. Value must be between 1 to 16	RO	4'b0000

4.3.11.10 cmn_hns_mpam_pri_idr

MPAM Priority partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1048

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-433: cmn_hns_mpam_pri_idr

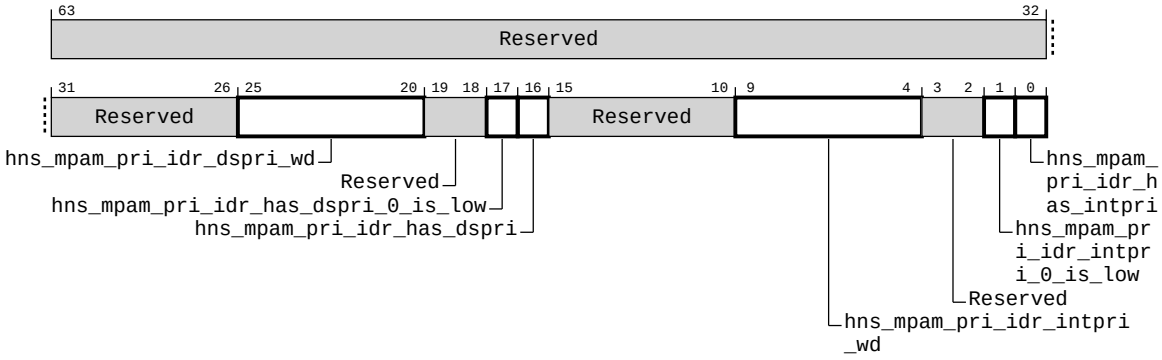


Table 4-449: cmn_hns_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	hns_mpam_pri_idr_dspri_wd	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	6'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_mpam_pri_idr_has_dspri_0_is_low	**0** In the DSPRI field, a value of 0 means highest priority **1** In the DSPRI field, a value of 0 means lowest priority.	RO	1'h0

Bits	Name	Description	Type	Reset
[16]	hns_mpam_pri_idr_has_dspri	**0** This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI. **1** This memory system component supports downstream priority and has an DSPRI field.	RO	1'h0
[15:10]	Reserved	Reserved	RO	-
[9:4]	hns_mpam_pri_idr_intpri_wd	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	6'h0
[3:2]	Reserved	Reserved	RO	-
[1]	hns_mpam_pri_idr_intpri_0_is_low	**0** In the INTPRI field, a value of 0 means highest priority. **1** In the INTPRI field, a value of 0 means lowest priority.	RO	1'h0
[0]	hns_mpam_pri_idr_has_intpri	**0** This memory system component supports priority, but doesn't have an internal priority field in MPAMCFG_PRI. **1** This memory system component supports internal priority and has an INTPRI field.	RO	1'h0

4.3.11.11 cmn_hns_mpam_partid_nrw_idr

MPAM PARTID narrowing ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1050`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-434: cmn_hns_mpam_partid_nrw_idr

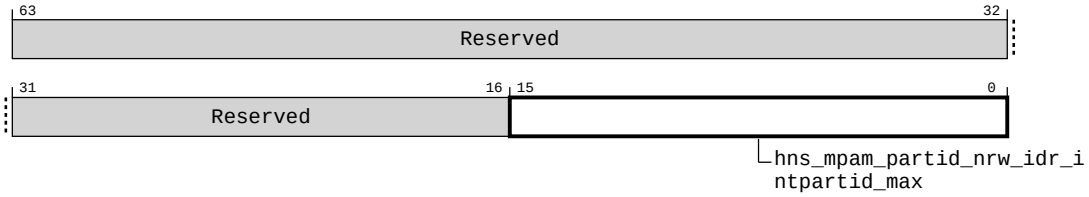


Table 4-450: cmn_hns_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

4.3.11.12 cmn_hns_mpam_msmon_idr

MPAM performance monitoring ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1080`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-435: cmn_hns_mpam_msmon_idr

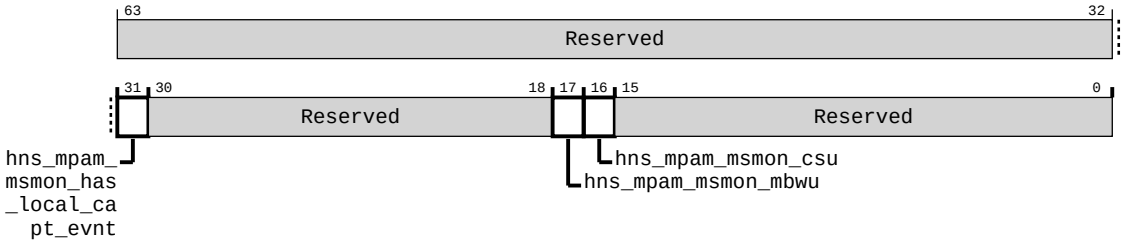


Table 4-451: cmn_hns_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_msmon_has_local_capt_evnt	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	1'h1
[30:18]	Reserved	Reserved	RO	-
[17]	hns_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	hns_mpam_msmon_csu	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

4.3.11.13 cmn_hns_mpam_csumon_idr

MPAM cache storage usage monitor ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1088`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-436: cmn_hns_mpam_csumon_idr

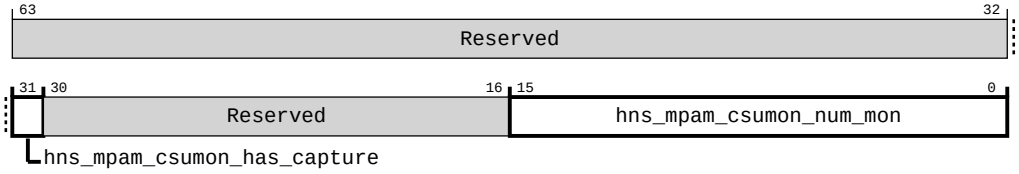


Table 4-452: cmn_hns_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_csumon_has_capture	**0** MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature **1** This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.	RO	1'h1
[30:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	Configuration dependent

4.3.11.14 cmn_hns_mpam_mbwumon_idr

MPAM memory bandwidth usage monitor ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1090`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-437: cmn_hns_mpam_mbwumon_idr

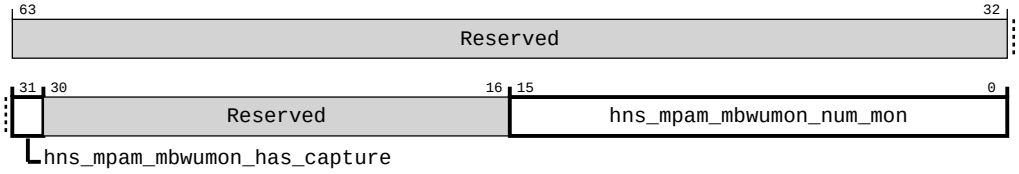


Table 4-453: cmn_hns_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_mbwumon_has_capture	**0** MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature. **1** This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.	RO	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	16'h0

4.3.11.15 cmn_hns_ns_mpam_ecr

MPAM Error Control Register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h10F0`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-438: cmn_hns_ns_mpam_ecr

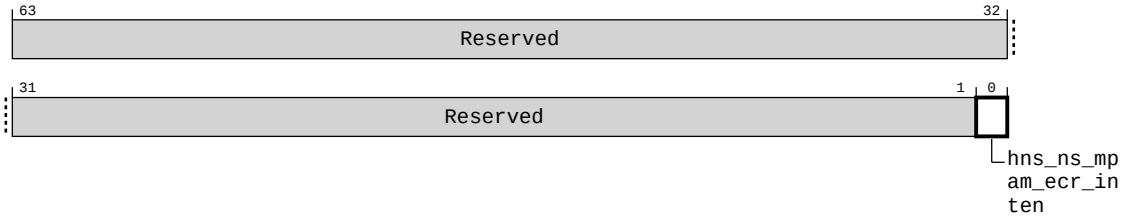


Table 4-454: cmn_hns_ns_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hns_ns_mpam_ecr_inten	Interrupt Enable. When INTEN = 0, MPAM error interrupts are not generated. When INTEN = 1, MPAM error interrupts are generated.	RW	1'h0

4.3.11.16 cmn_hns_ns_mpam_esr

MPAM Error Status Register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h10F8

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-439: cmn_hns_ns_mpam_esr

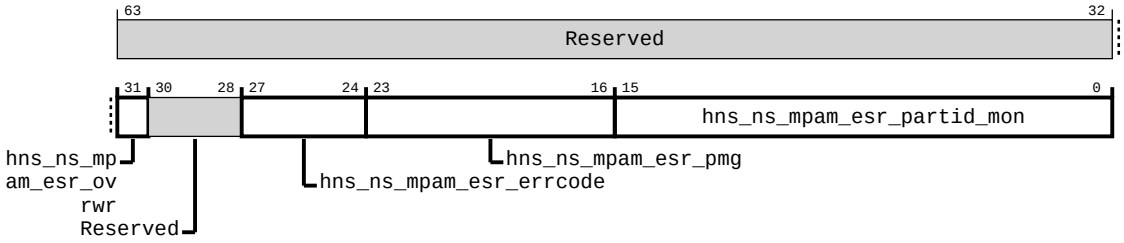


Table 4-455: cmn_hns_ns_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpam_esr_ovrwr	Overwritten. If 0 and ERRCODE is zero, no errors have occurred. If 0 and ERRCODE is non-zero, a single error has occurred and is recorded in this register. If 1 and ERRCODE is non-zero, multiple errors have occurred and this register records the most recent error. The state where this bit is 1 and ERRCODE is zero is not produced by hardware and is only reached when software writes this combination into this register.	RW	1'h0
[30:28]	Reserved	Reserved	RO	-
[27:24]	hns_ns_mpam_esr_errcode	Error code	RW	4'h0
[23:16]	hns_ns_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hns_ns_mpam_esr_partid_mon	PARTID captured if the error code captures PARTID. MON selector captured if the error code captures MON. Otherwise 0x0000.	RW	16'h0

4.3.11.17 cmn_hns_ns_mpamcfg_part_sel

MPAM partition configuration selection register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1100`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-440: cmn_hns_ns_mpamcfg_part_sel

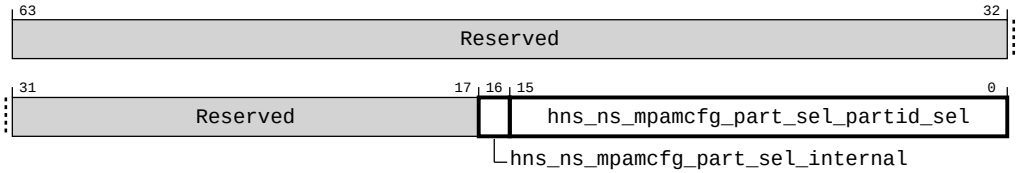


Table 4-456: cmn_hns_ns_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_ns_mpamcfg_part_sel_internal	If MPAMF_IDR.HAS_PARTID_NRW = 0, this field is RAZ/WI. If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interprete PARTID_SEL.	RW	1'h0
[15:0]	hns_ns_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

4.3.11.18 cmn_hns_ns_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1108`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-441: cmn_hns_ns_mpamcfg_cmax

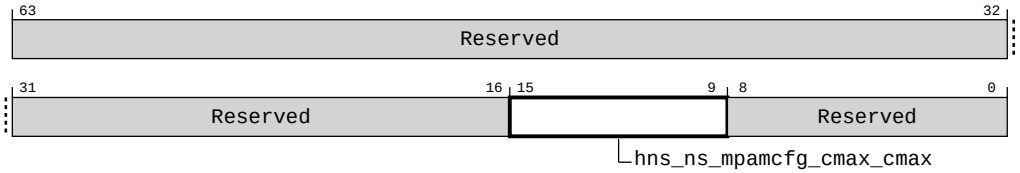


Table 4-457: cmn_hns_ns_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:9]	hns_ns_mpamcfg_cmax_cmax	Maximum cache capacity usage in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL.	RW	7'b1111111
[8:0]	Reserved	Reserved	RO	-

4.3.11.19 cmn_hns_ns_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1200`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-442: cmn_hns_ns_mpamcfg_mbw_min

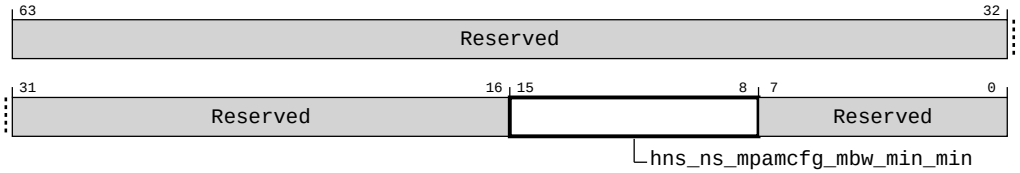


Table 4-458: cmn_hns_ns_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	hns_ns_mpamcfg_mbw_min_min	Memory minimum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	8'h0
[7:0]	Reserved	Reserved	RO	-

4.3.11.20 cmn_hns_ns_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register.

Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1208`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-443: cmn_hns_ns_mpamcfg_mbw_max

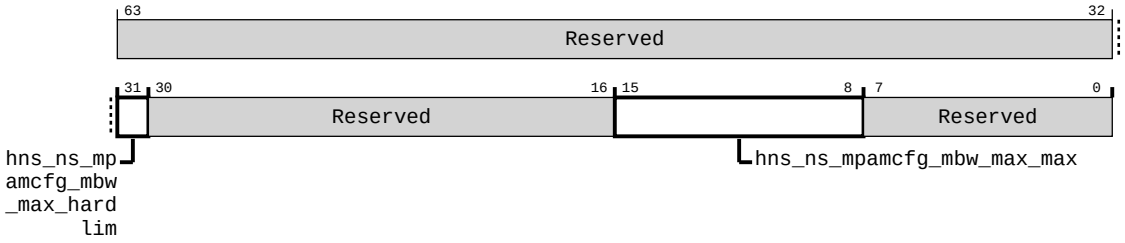


Table 4-459: cmn_hns_ns_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpamcfg_mbw_max_hardlim	**0** When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth **1** When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:8]	hns_ns_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	8'h0
[7:0]	Reserved	Reserved	RO	-

4.3.11.21 cmn_hns_ns_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register.

Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1220`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-444: cmn_hns_ns_mpamcfg_mbw_winwd

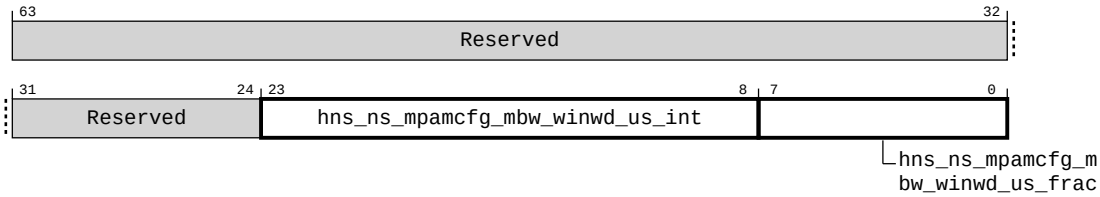


Table 4-460: cmn_hns_ns_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hns_ns_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hns_ns_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

4.3.11.22 cmn_hns_ns_mpamcfg_pri

MPAM priority partitioning configuration register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1400`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-445: cmn_hns_ns_mpamcfg_pri

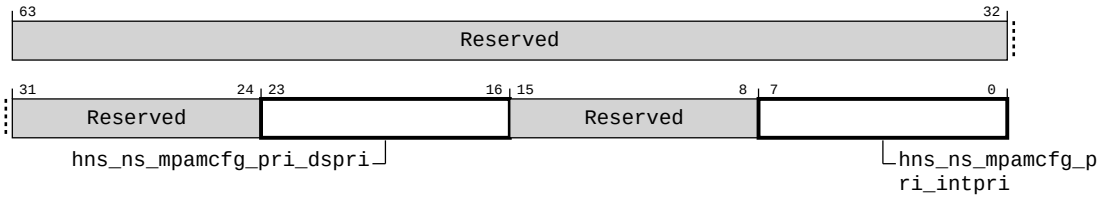


Table 4-461: cmn_hns_ns_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_ns_mpamcfg_pri_dspri	If HAS_DSPRI is 1, this field is a priority value applied to downstream communications from this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	8'h0
[15:8]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[7:0]	hns_ns_mpamcfg_pri_intpri	If HAS_INTPRI is 1, this field is a priority value applied internally inside this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	8'h0

4.3.11.23 cmn_hns_ns_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1500`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-446: cmn_hns_ns_mpamcfg_mbw_prop

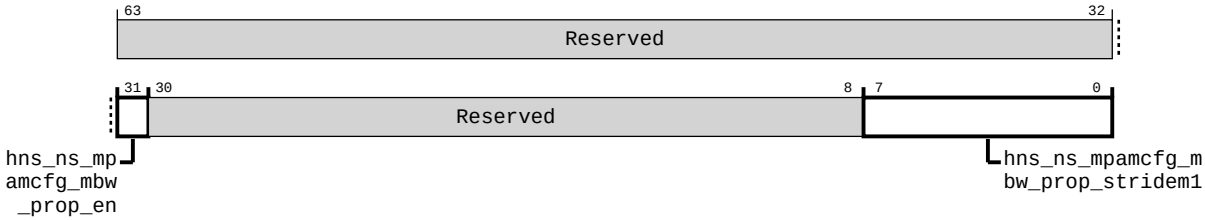


Table 4-462: cmn_hns_ns_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpamcfg_mbw_prop_en	**0** The selected partition is not regulated by proportional stride bandwidth partitioning. **1** The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.	RW	1'h0
[30:8]	Reserved	Reserved	RO	-
[7:0]	hns_ns_mpamcfg_mbw_prop_stridem1	Normalized cost of a bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	8'h0

4.3.11.24 cmn_hns_ns_mpamcfg_intpartid

MPAM internal partition narrowing configuration register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1600`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-447: cmn_hns_ns_mpamcfg_intpartid

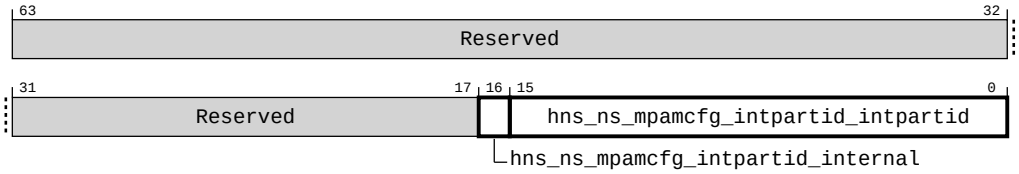


Table 4-463: cmn_hns_ns_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_ns_mpamcfg_intpartid_internal	This bit must be 1 when written to the register. If written as 0, the write will not update the reqPARTID to intPARTID association.	RW	1'h0
[15:0]	hns_ns_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

4.3.11.25 cmn_hns_ns_msmon_cfg_mon_sel

Memory system performance monitor selection register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1800`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-448: cmn_hns_ns_msmon_cfg_mon_sel

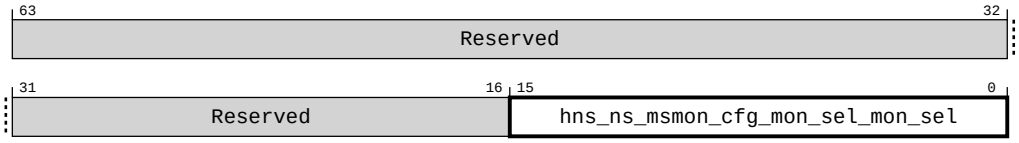


Table 4-464: cmn_hns_ns_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure.	RW	16'h0

4.3.11.26 cmn_hns_ns_msmon_capt_evnt

Memory system performance monitoring capture event generation register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1808`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-449: cmn_hns_ns_msmon_capt_evnt

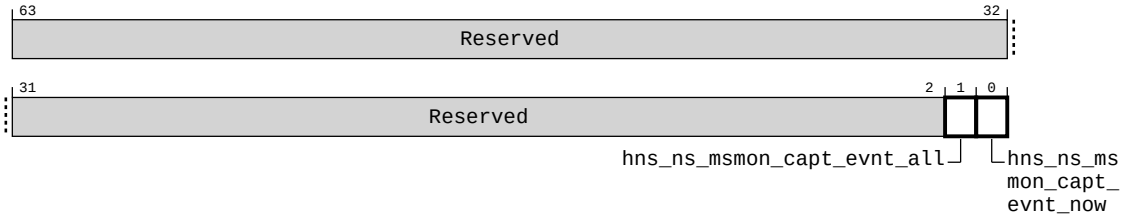


Table 4-465: cmn_hns_ns_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	hns_ns_msmon_capt_evnt_all	In Secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to Secure and Non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to Secure monitors in this memory system component with CAPT_EVNT = 7. In Non-secure version if NOW is written as 1, signal a capture event to Non-secure monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hns_ns_msmon_capt_evnt_now	When written as 1, this bit causes an event to all monitors in this memory system component with CAPT_EVNT set to the value of 7. When this bit is written as 0, no event is signalled.	RW	1'h0

4.3.11.27 cmn_hns_ns_msmon_cfg_csu_flt

Memory system performance monitor configure cache storage usage monitor filter register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1810`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-450: cmn_hns_ns_msmon_cfg_csu_flt

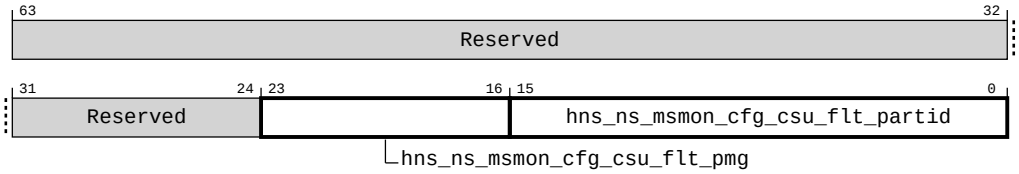


Table 4-466: cmn_hns_ns_msmon_cfg_csu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_ns_msmon_cfg_csu_flt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_ns_msmon_cfg_csu_flt_partid	Configures the cache storage usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	16'h0

4.3.11.28 cmn_hns_ns_msmon_cfg_csu_ctl

Memory system performance monitor configure cache storage usage monitor control register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1818

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-451: cmn_hns_ns_msmon_cfg_csu_ctl

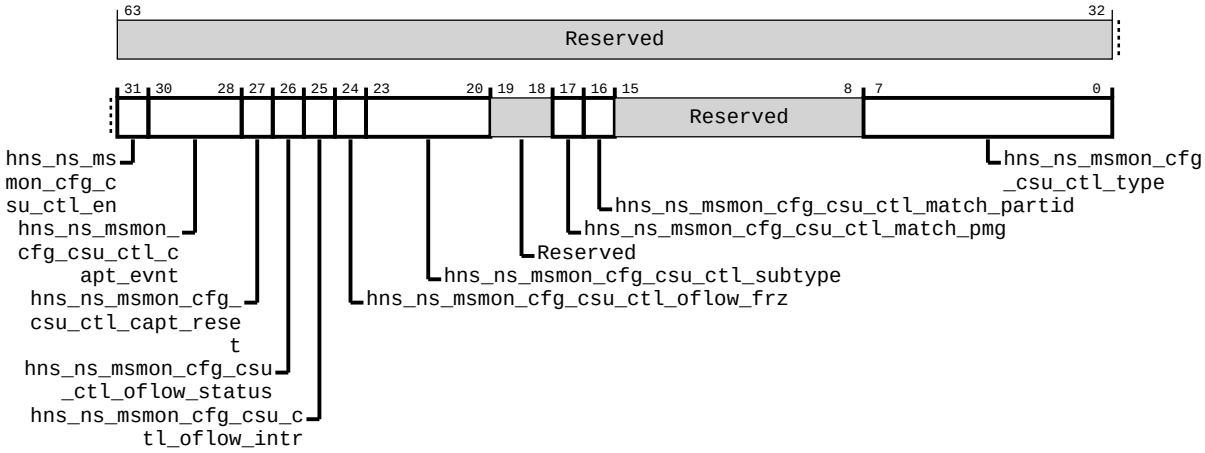


Table 4-467: cmn_hns_ns_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_cfg_csu_ctl_en	**0** The monitor is disabled and must not collect any information. **1** The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_ns_msmon_cfg_csu_ctl_capt_evnt	Select the event that triggers capture from the following: **0** No capture event is triggered **1** External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_ns_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	hns_ns_msmon_cfg_csu_ctl_oflow_status	**0** No overflow has occurred **1** At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_ns_msmon_cfg_csu_ctl_oflow_intr	**0** No interrupt. **1** On overflow, an implementation-specific interrupt is signalled.	RW	1'h0

Bits	Name	Description	Type	Reset
[24]	hns_ns_msmon_cfg_csu_ctl_oflow_frz	**0** Monitor count wraps on overflow. **1** Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1	RW	1'h0
[23:20]	hns_ns_msmon_cfg_csu_ctl_subtype	Not currently used for CSU monitors, but reserved for future use.	RW	4'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_ns_msmon_cfg_csu_ctl_match_pmg	**0** Monitor storage used by all PMG values. **1** Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_ns_msmon_cfg_csu_ctl_match_partid	**0** Monitor storage used by all PARTIDs. **1** Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_ns_msmon_cfg_csu_ctl_type	Read-only: Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	8'h43

4.3.11.29 cmn_hns_ns_msmon_cfg_mbwu_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register.


Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1820`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-452: cmn_hns_ns_msmon_cfg_mbwu_flt

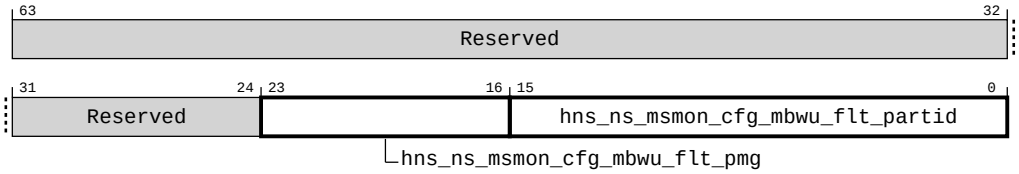


Table 4-468: cmn_hns_ns_msmon_cfg_mbwu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_ns_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_ns_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	16'h0

4.3.11.30 cmn_hns_ns_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1828`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-453: cmn_hns_ns_msmon_cfg_mbwu_ctl

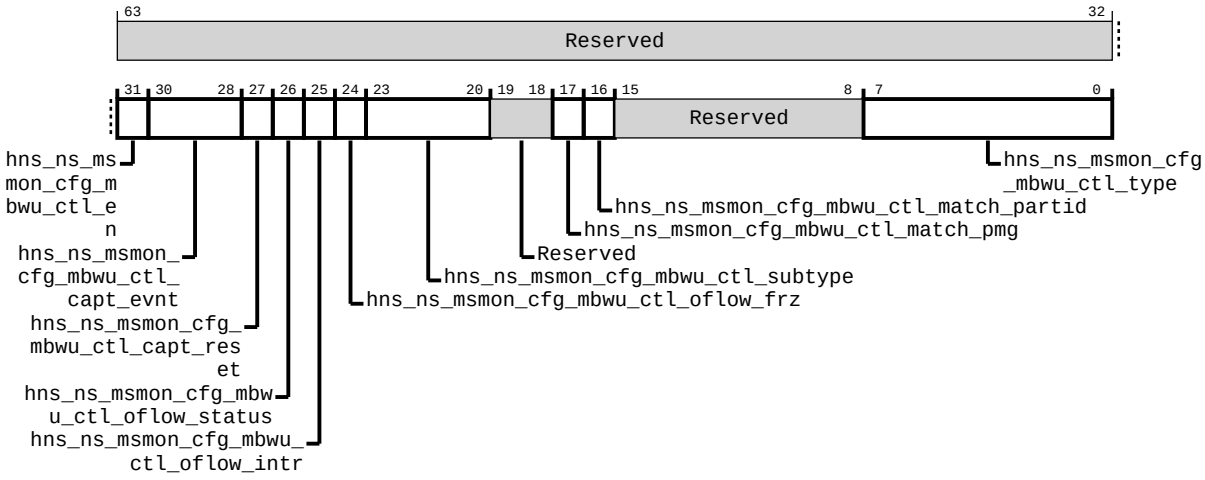


Table 4-469: cmn_hns_ns_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_cfg_mbwu_ctl_en	**0** The monitor is disabled and must not collect any information. **1** The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_ns_msmon_cfg_mbwu_ctl_capt_evnt	Select the event that triggers capture from the following: **0** No capture event is triggered **1** External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_ns_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hns_ns_msmon_cfg_mbwu_ctl_oflow_status	**0** No overflow has occurred **1** At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_ns_msmon_cfg_mbwu_ctl_oflow_intr	**0** No interrupt. **1** On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_ns_msmon_cfg_mbwu_ctl_oflow_frz	**0** Monitor count wraps on overflow. **1** Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1	RW	1'h0

Bits	Name	Description	Type	Reset
[23:20]	hns_ns_msmon_cfg_mbwu_ctl_subtype	A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports: **0** Do not count any bandwidth. **1** Count bandwidth used by memory reads **2** Count bandwidth used by memory writes **3** Count bandwidth used by memory reads and memory writes All other values are reserved and behaviour of a monitor with SUBTYPE set to one of the reserved values is UNPREDICTABLE.	RW	4'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_ns_msmon_cfg_mbwu_ctl_match_pmg	**0** Monitor bandwidth used by all PMG values. **1** Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_ns_msmon_cfg_mbwu_ctl_match_partid	**0** Monitor bandwidth used by all PARTIDs **1** Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_ns_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

4.3.11.31 cmn_hns_ns_msmon_csu

Memory system performance monitor cache storage usage monitor register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1840`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-454: cmn_hns_ns_msmon_csu

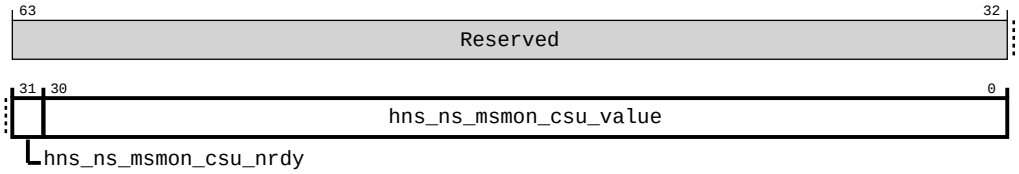


Table 4-470: cmn_hns_ns_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_csu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_csu_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

4.3.11.32 cmn_hns_ns_msmon_csu_capture

Memory system performance monitor cache storage usage capture register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1848

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-455: cmn_hns_ns_msmon_csu_capture

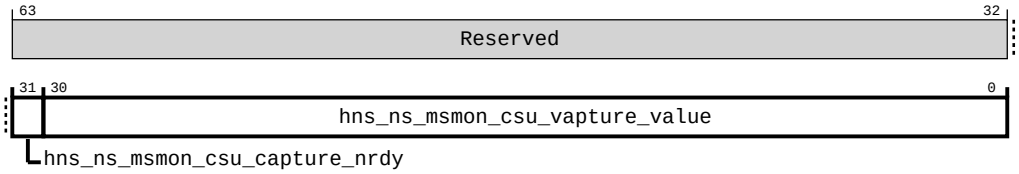


Table 4-471: cmn_hns_ns_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_csu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_csu_vapture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

4.3.11.33 cmn_hns_ns_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1860`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-456: cmn_hns_ns_msmon_mbwu

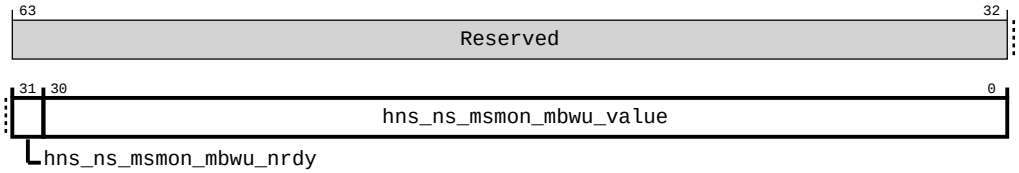


Table 4-472: cmn_hns_ns_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_mbwu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_mbwu_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

4.3.11.34 cmn_hns_ns_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register.

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1868

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-457: cmn_hns_ns_msmon_mbwu_capture

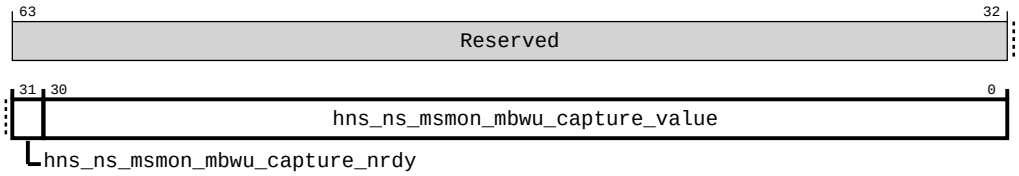


Table 4-473: cmn_hns_ns_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_mbwu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_mbwu_capture_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

4.3.11.35 cmn_hns_ns_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register.

Note

This register is unique for cmn_hns_ns. There is also similar but distinct register available in cmn_hns_s.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-458: cmn_hns_ns_mpamcfg_cpbm

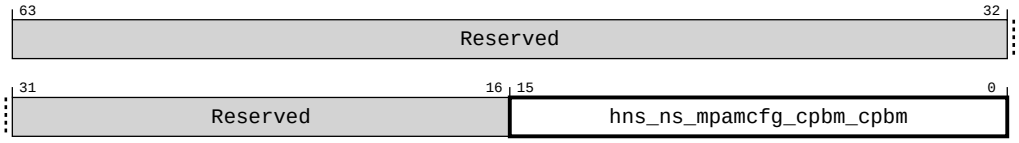


Table 4-474: cmn_hns_ns_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. NOTE: CPBM can not be all zeros for any PARTID.	RW	16'hFFFF

4.3.12 HN-F MPAM_S register descriptions

This section lists the HN-F MPAM_S registers.

4.3.12.1 cmn_hns_mpam_s_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-459: cmn_hns_mpam_s_node_info

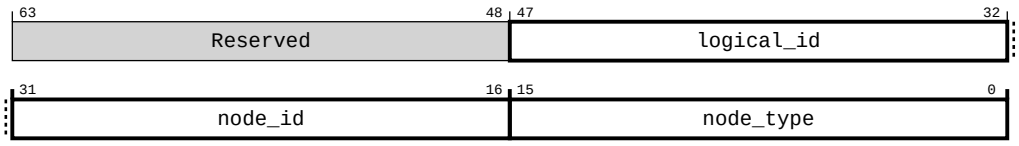


Table 4-475: cmn_hns_mpam_s_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	$logical_id_description	RO	Configuration dependent
[31:16]	node_id	$node_id_description	RO	Configuration dependent
[15:0]	node_type	$node_type_description	RO	Configuration dependent

4.3.12.2 cmn_hns_mpam_s_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h80

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-460: cmn_hns_mpam_s_child_info

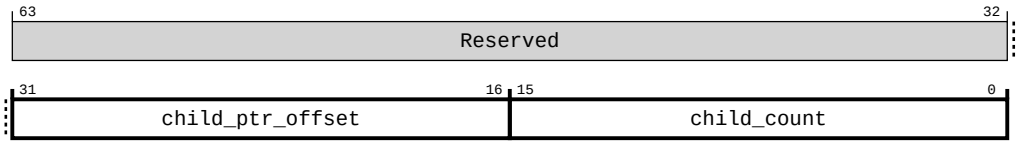


Table 4-476: cmn_hns_mpam_s_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.12.3 cmn_hns_mpam_s_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h980

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-461: cmn_hns_mpam_s_secure_register_groups_override

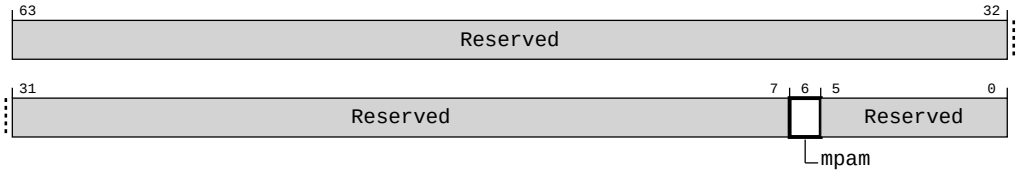


Table 4-477: cmn_hns_mpam_s_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	mpam	Allows Non-secure access to Secure MPAM registers	RW	1'b0
[5:0]	Reserved	Reserved	RO	-

4.3.12.4 cmn_hns_s_mpam_idr

MPAM features ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1000`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-462: cmn_hns_s_mpam_idr

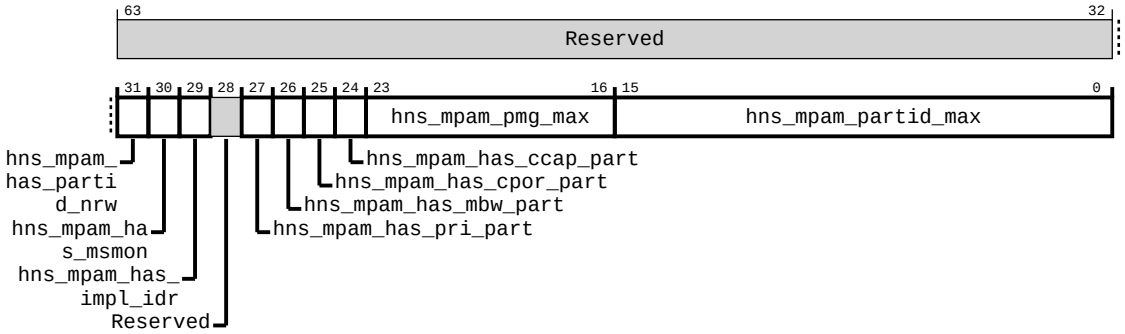


Table 4-478: cmn_hns_s_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_has_partid_nrw	**0** HN-F does not support MPAM PARTID Narrowing **1** HN-F supports MPAM PARTID Narrowing	RO	Configuration dependent
[30]	hns_mpam_has_msmon	**0** MPAM performance monitoring is not supported **1** MPAM performance monitoring is supported	RO	Configuration dependent
[29]	hns_mpam_has_impl_idr	**0** MPAM implementation specific partitioning features not supported **1** MPAM implementation specific partitioning features supported	RO	Configuration dependent
[28]	Reserved	Reserved	RO	-
[27]	hns_mpam_has_pri_part	**0** MPAM priority partitioning is not supported **1** MPAM priority partitioning is supported	RO	Configuration dependent
[26]	hns_mpam_has_mbw_part	**0** MPAM memory bandwidth partitioning is not supported **1** MPAM memory bandwidth partitioning is supported	RO	Configuration dependent
[25]	hns_mpam_has_cpor_part	**0** MPAM cache portion partitioning is not supported **1** MPAM cache portion partitioning is supported	RO	Configuration dependent
[24]	hns_mpam_has_ccap_part	**0** MPAM cache maximum capacity partitioning is not supported **1** MPAM cache maximum capacity partitioning is supported	RO	Configuration dependent
[23:16]	hns_mpam_pmg_max	Maximum value of Non-secure PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_mpam_partid_max	Maximum value of Non-secure PARTID supported by this HN-F	RO	Configuration dependent

4.3.12.5 cmn_hns_mpam_sidr

MPAM features Secure ID register. This is Secure (S) register only.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1008`

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-463: cmn_hns_mpam_sidr

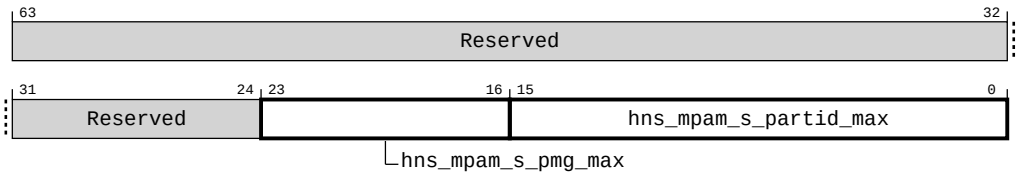


Table 4-479: cmn_hns_mpam_sidr attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_mpam_s_pmg_max	Maximum value of Secure PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_mpam_s_partid_max	Maximum value of Secure PARTID supported by this HN-F	RO	Configuration dependent

4.3.12.6 cmn_hns_s_mpam_iidr

MPAM Implementation ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1018`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-464: cmn_hns_s_mpam_iidr

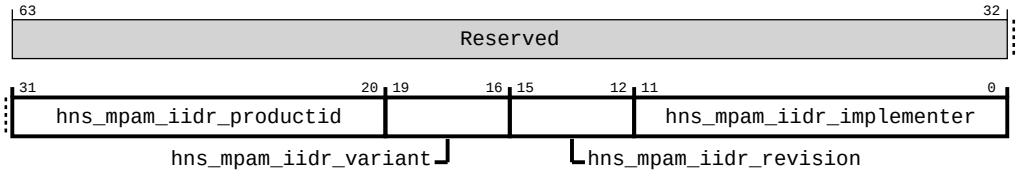


Table 4-480: cmn_hns_s_mpam_iidr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hns_mpam_iidr_productid	Implementation defined value identifying MPAM memory system component	RO	12'h000
[19:16]	hns_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	4'b0000
[15:12]	hns_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	4'b0000
[11:0]	hns_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	12'h43B

4.3.12.7 cmn_hns_s_mpam_aidr

MPAM architecture ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1020`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-465: cmn_hns_s_mpam_aidr

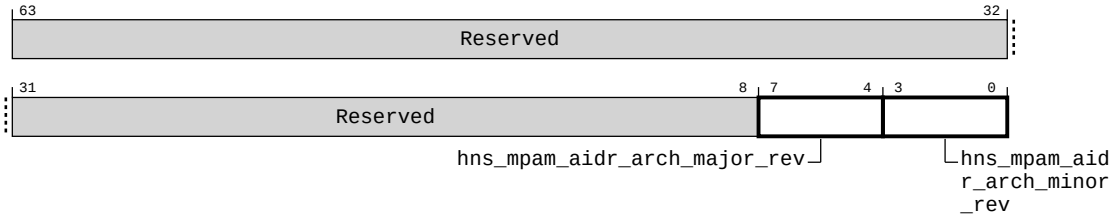


Table 4-481: cmn_hns_s_mpam_aidr attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hns_mpam_aidr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	4'b0001
[3:0]	hns_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	4'b0000

4.3.12.8 cmn_hns_s_mpam_impl_idr

MPAM Implementation defined partitioning feature ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1028`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-466: cmn_hns_s_mpam_impl_idr

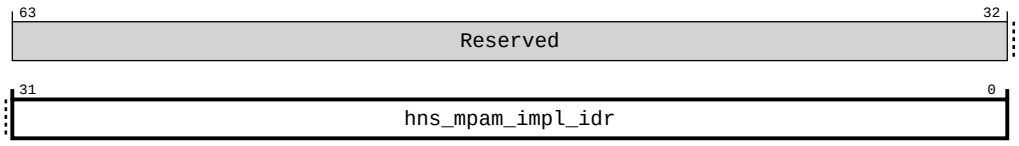


Table 4-482: cmn_hns_s_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hns_mpam_impl_idr	Implementation defined partitioning features.	RO	32'h00000000

4.3.12.9 cmn_hns_s_mpam_cpor_idr

MPAM cache portion partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1030`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-467: cmn_hns_s_mpam_cpor_idr

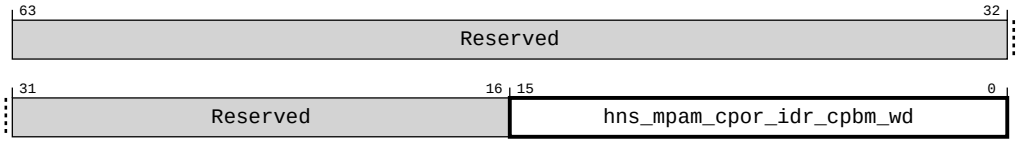


Table 4-483: cmn_hns_s_mpam_cpor_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bit map of this device.	RO	Configuration dependent

4.3.12.10 cmn_hns_s_mpam_ccap_idr

MPAM cache capacity partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1038`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-468: cmn_hns_s_mpam_ccap_idr

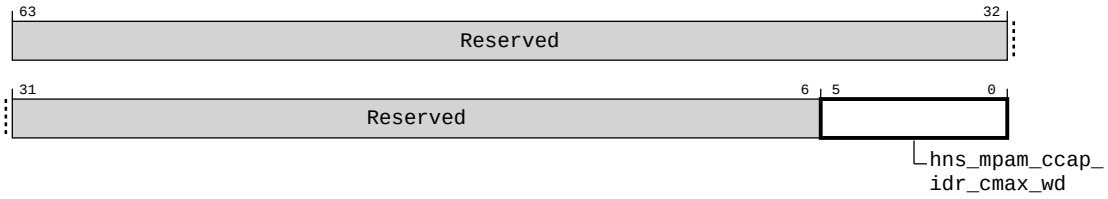


Table 4-484: cmn_hns_s_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:0]	hns_mpam_ccap_idr_cmax_wd	Number of fractional bits implemented in the cache capacity partitioning.	RO	Configuration dependent

4.3.12.11 cmn_hns_s_mpam_mbw_idr

MPAM Memory Bandwidth partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1040`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-469: cmn_hns_s_mpam_mbw_idr

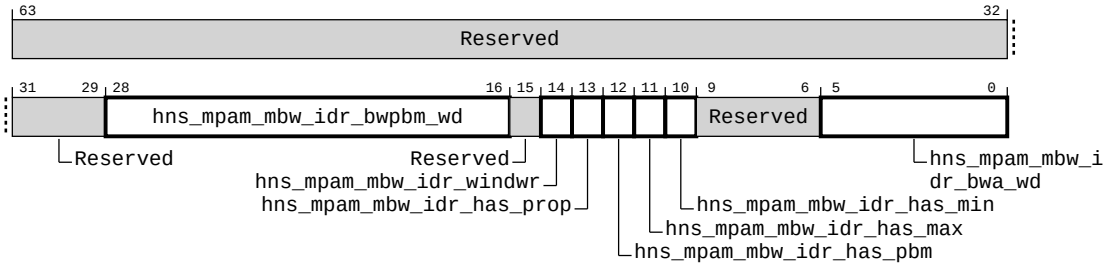


Table 4-485: cmn_hns_s_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hns_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register.	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hns_mpam_mbw_idr_windwr	**0** The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed **1** The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.	RO	1'h0
[13]	hns_mpam_mbw_idr_has_prop	**0** There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register **1** MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.	RO	1'h0
[12]	hns_mpam_mbw_idr_has_pbm	**0** There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register **1** MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.	RO	1'h0
[11]	hns_mpam_mbw_idr_has_max	**0** There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register **1** MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.	RO	1'h0
[10]	hns_mpam_mbw_idr_has_min	**0** There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register **1** MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.	RO	1'h0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_mpam_mbw_idr_bwa_wd	Number of implemented bits in bandwidth allocation fields: MIN, MAX, and STRIDE. Value must be between 1 to 16	RO	4'b0000

4.3.12.12 cmn_hns_s_mpam_pri_idr

MPAM Priority partitioning ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1048`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-470: cmn_hns_s_mpam_pri_idr

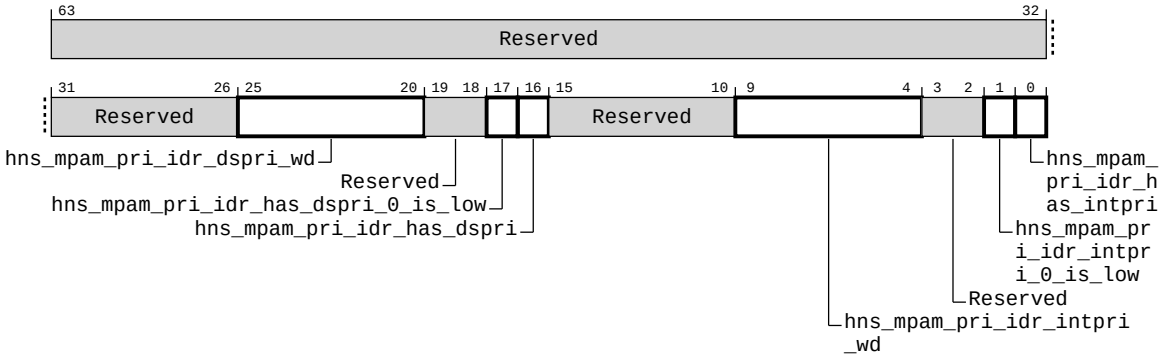


Table 4-486: cmn_hns_s_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	hns_mpam_pri_idr_dspri_wd	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	6'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_mpam_pri_idr_has_dspri_0_is_low	**0** In the DSPRI field, a value of 0 means highest priority **1** In the DSPRI field, a value of 0 means lowest priority.	RO	1'h0
[16]	hns_mpam_pri_idr_has_dspri	**0** This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI. **1** This memory system component supports downstream priority and has an DSPRI field.	RO	1'h0
[15:10]	Reserved	Reserved	RO	-
[9:4]	hns_mpam_pri_idr_intpri_wd	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	6'h0
[3:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1]	hns_mpam_pri_idr_intpri_0_is_low	**0** In the INTPRI field, a value of 0 means highest priority. **1** In the INTPRI field, a value of 0 means lowest priority.	RO	1'h0
[0]	hns_mpam_pri_idr_has_intpri	**0** This memory system component supports priority, but doesn't have an internal priority field in MPAMCFG_PRI. **1** This memory system component supports internal priority and has an INTPRI field.	RO	1'h0

4.3.12.13 cmn_hns_s_mpam_partid_nrw_idr

MPAM PARTID narrowing ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1050`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-471: cmn_hns_s_mpam_partid_nrw_idr

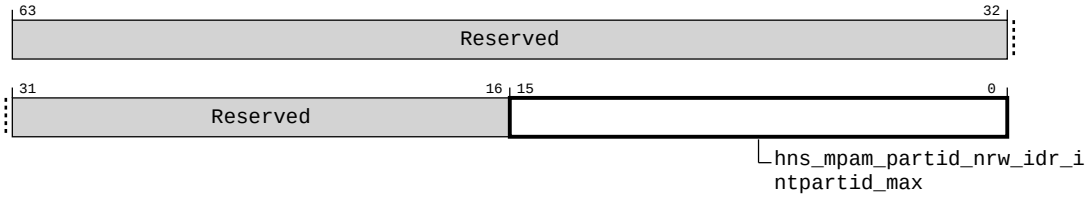


Table 4-487: cmn_hns_s_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

4.3.12.14 cmn_hns_s_mpam_msmon_idr

MPAM performance monitoring ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1080

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-472: cmn_hns_s_mpam_msmon_idr

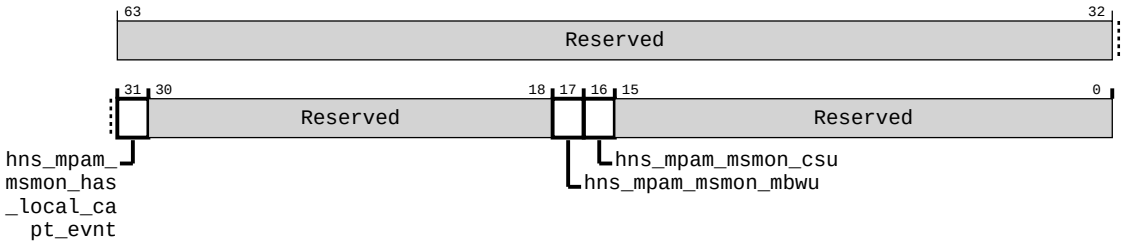


Table 4-488: cmn_hns_s_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_msmon_has_local_capt_evnt	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	1'h1
[30:18]	Reserved	Reserved	RO	-
[17]	hns_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	hns_mpam_msmon_csu	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[15:0]	Reserved	Reserved	RO	-

4.3.12.15 cmn_hns_s_mpam_csumon_idr

MPAM cache storage usage monitor ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1088`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-473: cmn_hns_s_mpam_csumon_idr

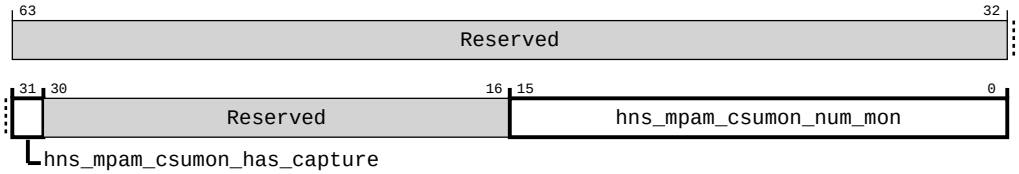


Table 4-489: cmn_hns_s_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_csumon_has_capture	**0** MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature **1** This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.	RO	1'h1
[30:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	Configuration dependent

4.3.12.16 cmn_hns_s_mpam_mbwumon_idr

MPAM memory bandwidth usage monitor ID register. This is a shared register for S and NS

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1090`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-474: cmn_hns_s_mpam_mbwumon_idr

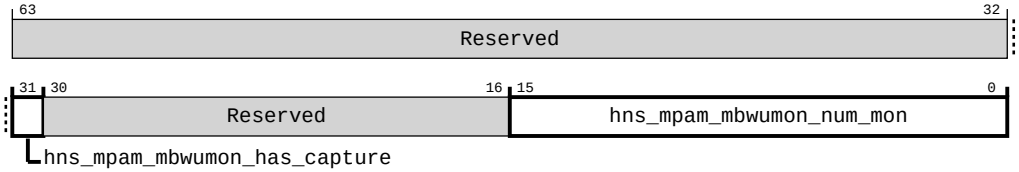


Table 4-490: cmn_hns_s_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_mpam_mbwumon_has_capture	**0** MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature. **1** This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.	RO	1'h0

Bits	Name	Description	Type	Reset
[30:16]	Reserved	Reserved	RO	-
[15:0]	hns_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	16'h0

4.3.12.17 cmn_hns_s_mpam_ecr

MPAM Error Control Register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h10F0`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-475: cmn_hns_s_mpam_ecr

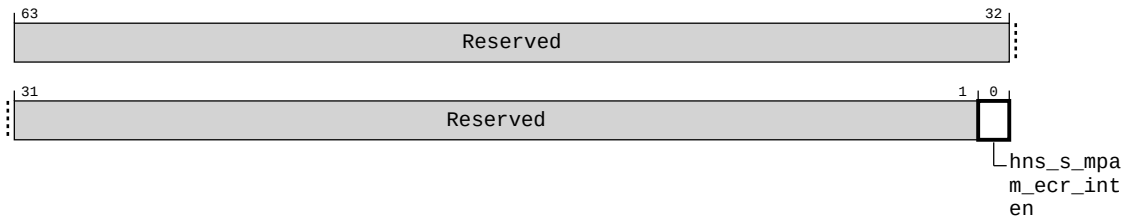


Table 4-491: cmn_hns_s_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hns_s_mpam_ecr_inten	Interrupt Enable. When INTEN = 0, MPAM error interrupts are not generated. When INTEN = 1, MPAM error interrupts are generated.	RW	1'h0

4.3.12.18 cmn_hns_s_mpam_esr

MPAM Error Status Register.


Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h10F8`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-476: cmn_hns_s_mpam_esr

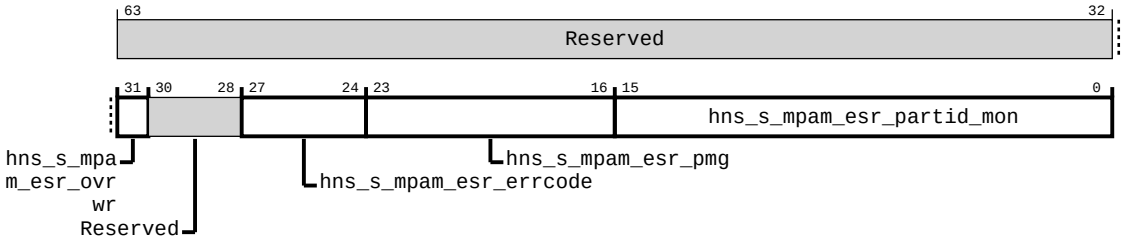


Table 4-492: cmn_hns_s_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_mpam_esr_ovrwr	Overwritten. If 0 and ERRCODE is zero, no errors have occurred. If 0 and ERRCODE is non-zero, a single error has occurred and is recorded in this register. If 1 and ERRCODE is non-zero, multiple errors have occurred and this register records the most recent error. The state where this bit is 1 and ERRCODE is zero is not produced by hardware and is only reached when software writes this combination into this register.	RW	1'h0
[30:28]	Reserved	Reserved	RO	-
[27:24]	hns_s_mpam_esr_errcode	Error code	RW	4'h0
[23:16]	hns_s_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hns_s_mpam_esr_partid_mon	PARTID captured if the error code captures PARTID. MON selector captured if the error code captures MON. Otherwise 0x0000.	RW	16'h0

4.3.12.19 cmn_hns_s_mpamcfg_part_sel

MPAM partition configuration selection register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1100`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-477: cmn_hns_s_mpamcfg_part_sel

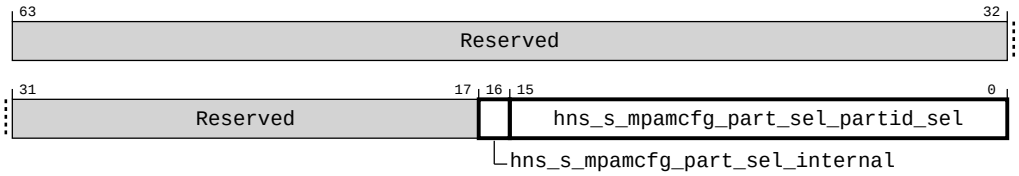


Table 4-493: cmn_hns_s_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_s_mpamcfg_part_sel_internal	If MPAMF_IDR.HAS_PARTID_NRW = 0, this field is RAZ/WI. If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interprete PARTID_SEL.	RW	1'h0
[15:0]	hns_s_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

4.3.12.20 cmn_hns_s_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register.

Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1108`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-478: cmn_hns_s_mpamcfg_cmax

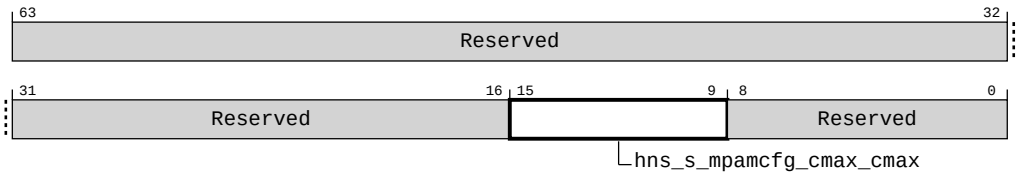


Table 4-494: cmn_hns_s_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:9]	hns_s_mpamcfg_cmax_cmax	Maximum cache capacity usage in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL.	RW	7'b1111111
[8:0]	Reserved	Reserved	RO	-

4.3.12.21 cmn_hns_s_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register.

Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1200`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-479: cmn_hns_s_mpamcfg_mbw_min

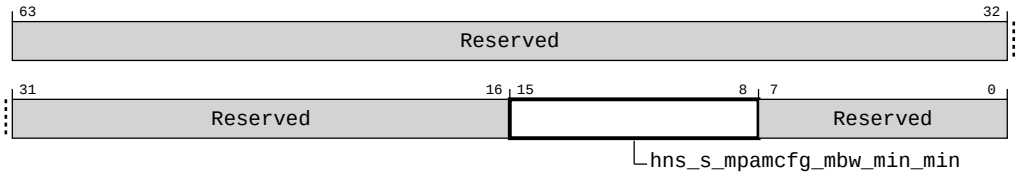


Table 4-495: cmn_hns_s_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	hns_s_mpamcfg_mbw_min_min	Memory minimum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	8'h0

Bits	Name	Description	Type	Reset
[7:0]	Reserved	Reserved	RO	-

4.3.12.22 cmn_hns_s_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1208`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-480: cmn_hns_s_mpamcfg_mbw_max

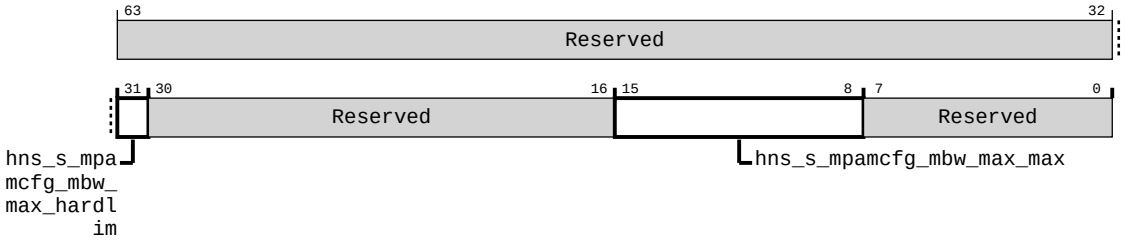


Table 4-496: cmn_hns_s_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_mpamcfg_mbw_max_hardlim	**0** When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth **1** When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:8]	hns_s_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	8'h0
[7:0]	Reserved	Reserved	RO	-

4.3.12.23 cmn_hns_s_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register.


Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1220`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-481: cmn_hns_s_mpamcfg_mbw_winwd

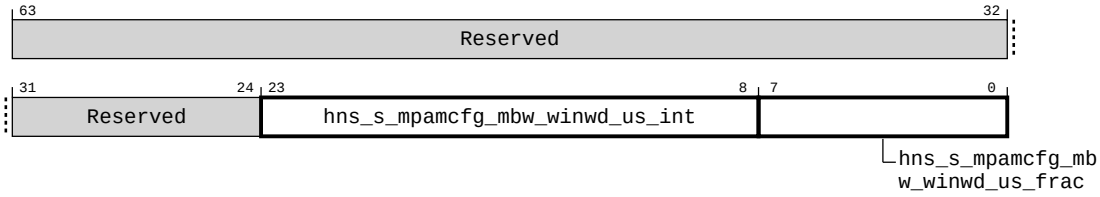


Table 4-497: cmn_hns_s_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hns_s_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hns_s_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

4.3.12.24 cmn_hns_s_mpamcfg_pri

MPAM priority partitioning configuration register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1400`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-482: cmn_hns_s_mpamcfg_pri

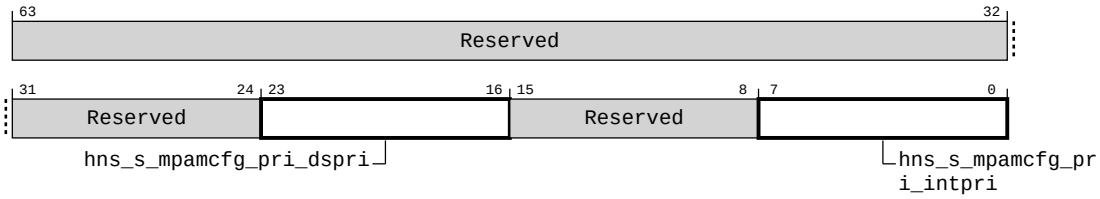


Table 4-498: cmn_hns_s_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_s_mpamcfg_pri_dspri	If HAS_DSPRI is 1, this field is a priority value applied to downstream communications from this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	8'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_s_mpamcfg_pri_intpri	If HAS_INTPRI is 1, this field is a priority value applied internally inside this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	8'h0

4.3.12.25 cmn_hns_s_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1500`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-483: cmn_hns_s_mpamcfg_mbw_prop

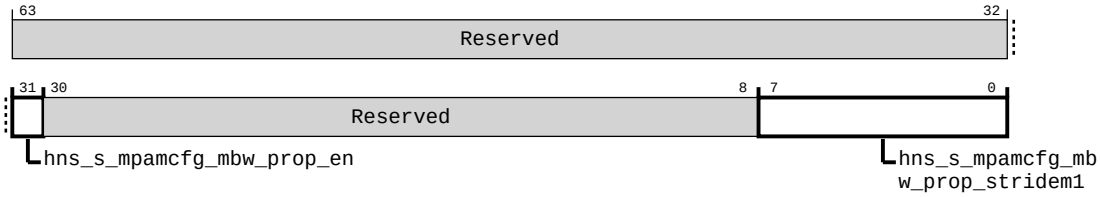


Table 4-499: cmn_hns_s_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_mpamcfg_mbw_prop_en	**0** The selected partition is not regulated by proportional stride bandwidth partitioning. **1** The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.	RW	1'h0
[30:8]	Reserved	Reserved	RO	-
[7:0]	hns_s_mpamcfg_mbw_prop_stridem1	Normalized cost of a bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	8'h0

4.3.12.26 cmn_hns_s_mpamcfg_intpartid

MPAM internal partition narrowing configuration register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1600

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-484: cmn_hns_s_mpamcfg_intpartid

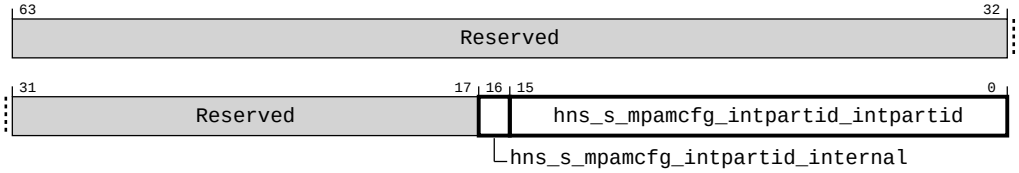


Table 4-500: cmn_hns_s_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_s_mpamcfg_intpartid_internal	This bit must be 1 when written to the register. If written as 0, the write will not update the reqPARTID to intPARTID association.	RW	1'h0
[15:0]	hns_s_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

4.3.12.27 cmn_hns_s_msmon_cfg_mon_sel

Memory system performance monitor selection register.


Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1800`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-485: cmn_hns_s_msmon_cfg_mon_sel

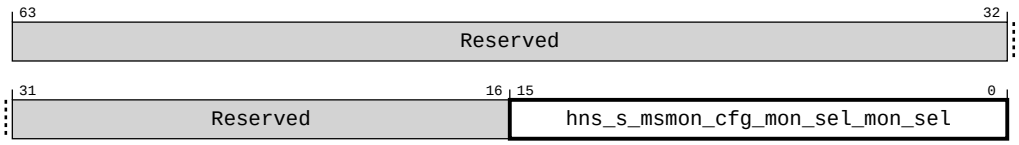


Table 4-501: cmn_hns_s_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure.	RW	16'h0

4.3.12.28 cmn_hns_s_msmon_capt_evnt

Memory system performance monitoring capture event generation register.

Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1808

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-486: cmn_hns_s_msmon_capt_evnt

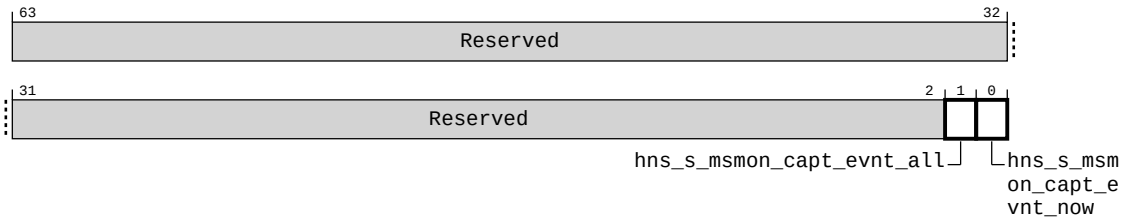


Table 4-502: cmn_hns_s_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	hns_s_msmon_capt_evnt_all	In Secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to Secure and Non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to Secure monitors in this memory system component with CAPT_EVNT = 7. In Non-secure version if NOW is written as 1, signal a capture event to Non-secure monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hns_s_msmon_capt_evnt_now	When written as 1, this bit causes an event to all monitors in this memory system component with CAPT_EVNT set to the value of 7. When this bit is written as 0, no event is signalled.	RW	1'h0

4.3.12.29 cmn_hns_s_msmon_cfg_csu_flt

Memory system performance monitor configure cache storage usage monitor filter register.

Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1810

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-487: cmn_hns_s_msmon_cfg_csu_flt

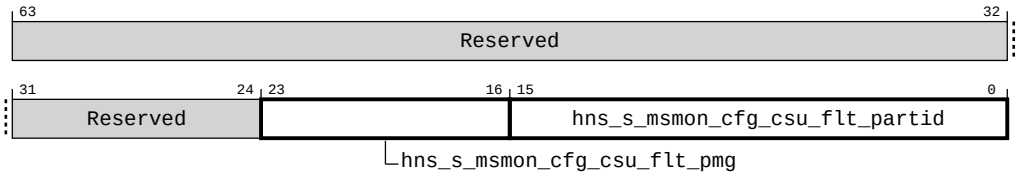


Table 4-503: cmn_hns_s_msmon_cfg_csu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[23:16]	hns_s_msmon_cfg_csu_flt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_s_msmon_cfg_csu_flt_partid	Configures the cache storage usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	16'h0

4.3.12.30 cmn_hns_s_msmon_cfg_csu_ctl

Memory system performance monitor configure cache storage usage monitor control register.


Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1818`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-488: cmn_hns_s_msmon_cfg_csu_ctl

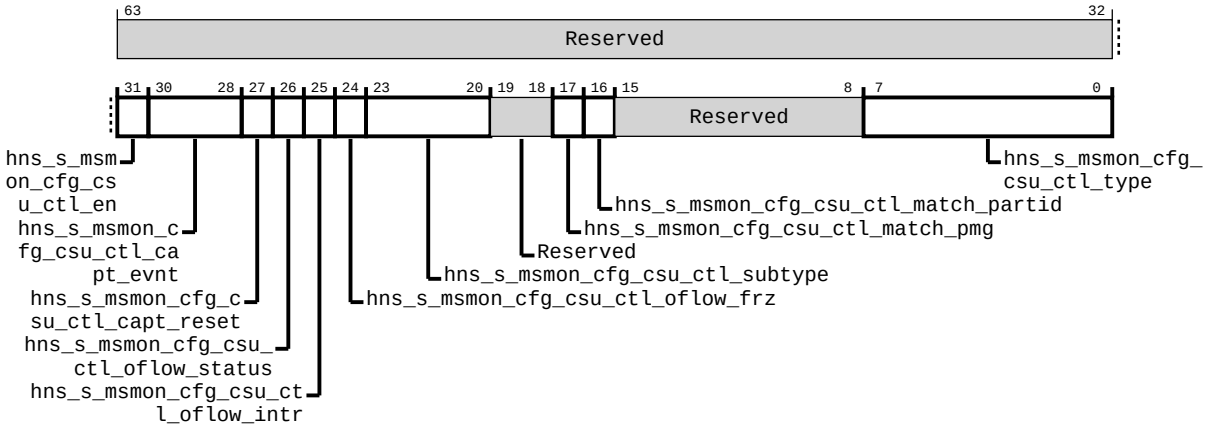


Table 4-504: cmn_hns_s_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_cfg_csu_ctl_en	**0** The monitor is disabled and must not collect any information. **1** The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_s_msmon_cfg_csu_ctl_capt_evnt	Select the event that triggers capture from the following: **0** No capture event is triggered **1** External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_s_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	hns_s_msmon_cfg_csu_ctl_oflow_status	**0** No overflow has occurred **1** At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_s_msmon_cfg_csu_ctl_oflow_intr	**0** No interrupt. **1** On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_s_msmon_cfg_csu_ctl_oflow_frz	**0** Monitor count wraps on overflow. **1** Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1	RW	1'h0
[23:20]	hns_s_msmon_cfg_csu_ctl_subtype	Not currently used for CSU monitors, but reserved for future use.	RW	4'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_s_msmon_cfg_csu_ctl_match_pmg	**0** Monitor storage used by all PMG values. **1** Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_s_msmon_cfg_csu_ctl_match_partid	**0** Monitor storage used by all PARTIDs. **1** Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_s_msmon_cfg_csu_ctl_type	Read-only: Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	8'h43

4.3.12.31 cmn_hns_s_msmon_cfg_mbwu_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1820`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-489: cmn_hns_s_msmon_cfg_mbwu_flt

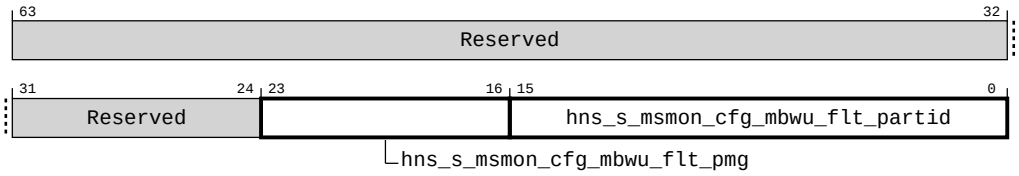


Table 4-505: cmn_hns_s_msmon_cfg_mbwu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[23:16]	hns_s_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_s_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	16'h0

4.3.12.32 cmn_hns_s_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.


Note

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1828`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-490: cmn_hns_s_msmon_cfg_mbwu_ctl

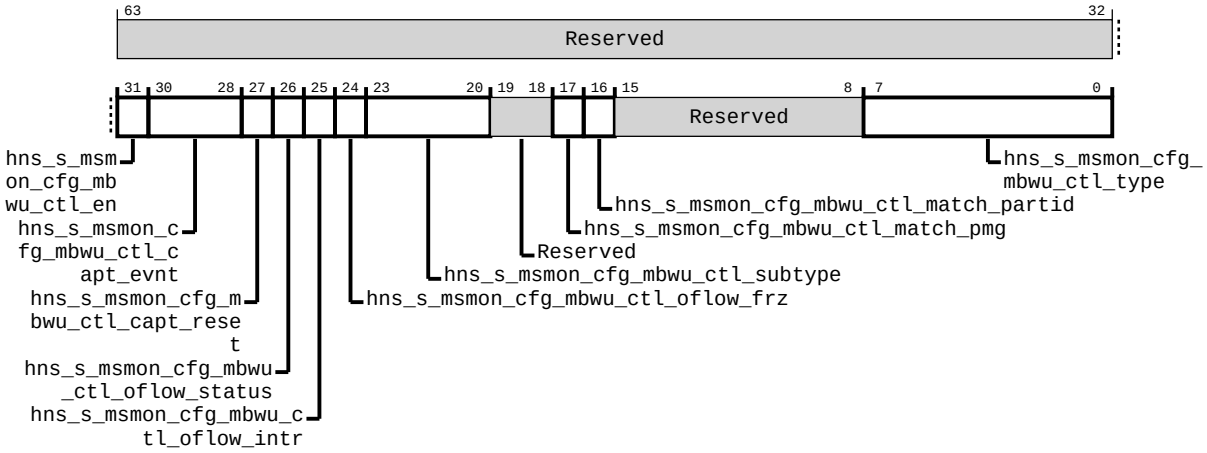


Table 4-506: cmn_hns_s_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_cfg_mbwu_ctl_en	**0** The monitor is disabled and must not collect any information. **1** The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_s_msmon_cfg_mbwu_ctl_capt_evnt	Select the event that triggers capture from the following: **0** No capture event is triggered **1** External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_s_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hns_s_msmon_cfg_mbwu_ctl_oflow_status	**0** No overflow has occurred **1** At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_s_msmon_cfg_mbwu_ctl_oflow_intr	**0** No interrupt. **1** On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_s_msmon_cfg_mbwu_ctl_oflow_frz	**0** Monitor count wraps on overflow. **1** Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1	RW	1'h0
[23:20]	hns_s_msmon_cfg_mbwu_ctl_subtype	A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports: **0** Do not count any bandwidth. **1** Count bandwidth used by memory reads **2** Count bandwidth used by memory writes **3** Count bandwidth used by memory reads and memory writes All other values are reserved and behaviour of a monitor with SUBTYPE set to one of the reserved values is UNPREDICTABLE.	RW	4'h0
[19:18]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[17]	hns_s_msmon_cfg_mbwu_ctl_match_pmg	**0** Monitor bandwidth used by all PMG values. **1** Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_s_msmon_cfg_mbwu_ctl_match_partid	**0** Monitor bandwidth used by all PARTIDs **1** Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_s_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

4.3.12.33 cmn_hns_s_msmon_csu

Memory system performance monitor cache storage usage monitor register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1840`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-491: cmn_hns_s_msmon_csu

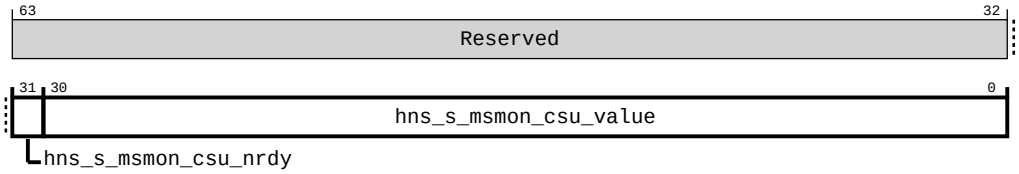


Table 4-507: cmn_hns_s_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_csu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_csu_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

4.3.12.34 cmn_hns_s_msmon_csu_capture

Memory system performance monitor cache storage usage capture register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1848`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-492: cmn_hns_s_msmon_csu_capture

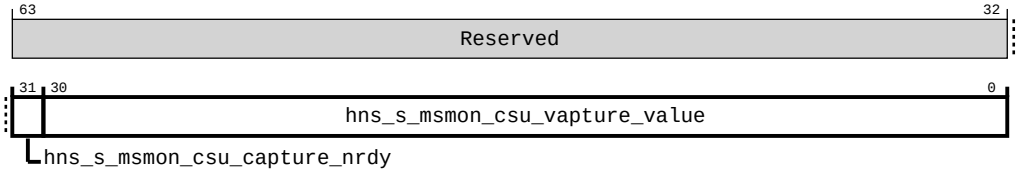


Table 4-508: cmn_hns_s_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_csu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_csu_vapture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

4.3.12.35 cmn_hns_s_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1860`

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-493: cmn_hns_s_msmon_mbwu

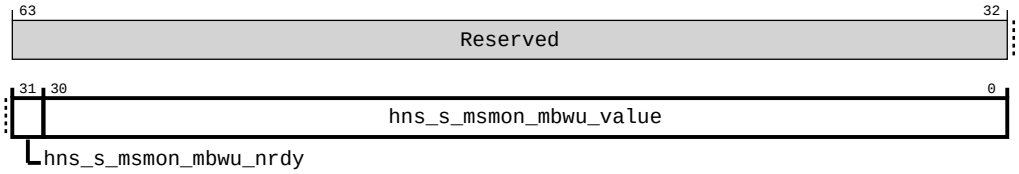


Table 4-509: cmn_hns_s_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_mbwu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_mbwu_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

4.3.12.36 cmn_hns_s_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1868

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-494: cmn_hns_s_msmon_mbwu_capture

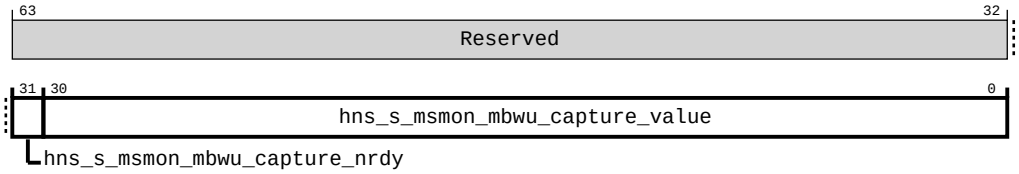


Table 4-510: cmn_hns_s_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_mbwu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_mbwu_capture_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

4.3.12.37 cmn_hns_s_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register.

This register is unique for cmn_hns_s. There is also similar but distinct register available in cmn_hns_ns.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2000

Type

RW

Reset value

See individual bit resets

Secure group override

cmn_hns_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-495: cmn_hns_s_mpamcfg_cpbm

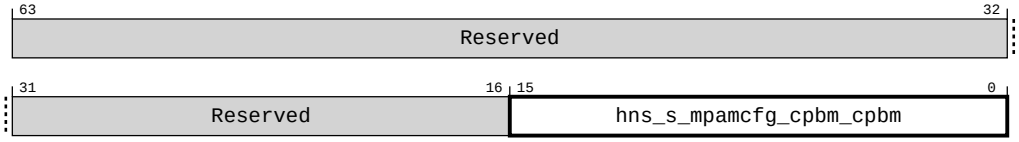


Table 4-511: cmn_hns_s_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. NOTE: CPBM can not be all zeros for any PARTID.	RW	16'hFFFF

4.3.13 MXP register descriptions

This section lists the MXP registers.

4.3.13.1 por_mxp_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-496: por_mxp_node_info

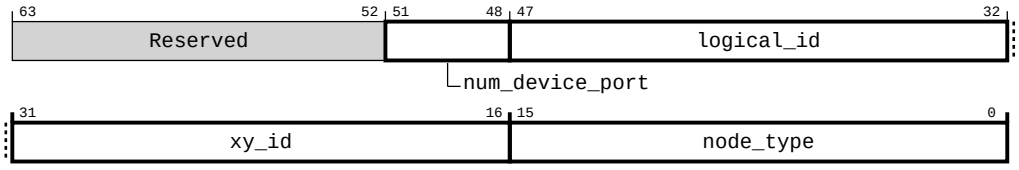


Table 4-512: por_mxp_node_info attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:48]	num_device_port	Number of device ports attached to the MXP. Mesh config = (1x1)? Max. of 6 Device Ports are supported : Max. of 4 Device Ports are supported	RO	Configuration dependent
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	xy_id	Identifies (X,Y) location of XP within the mesh NOTE: The (X,Y) location is specified following the node ID format as defined in Node ID mapping section, with the bottom 3 bits, corresponding to port ID and device ID, set to 0. Bits 31:11 must always be set to 0. The range of bits representing the (X,Y) location varies for different node ID formats.	RO	16'h0000
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0006

4.3.13.2 por_mxp_device_port_connect_info_p0-5

There are 6 iterations of this register. The index ranges from 0 to 5. Contains device port connection information for port #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h8 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-497: por_mxp_device_port_connect_info_p0-5

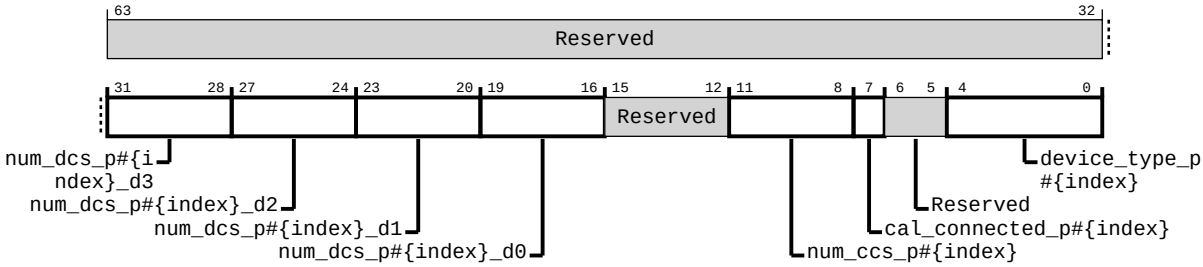


Table 4-513: por_mxp_device_port_connect_info_p0-5 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	num_dcs_p#{index}_d3	Number of device credited slices connected to port #{index} device 3 (Allowed values: 0-4)	RO	Configuration dependent
[27:24]	num_dcs_p#{index}_d2	Number of device credited slices connected to port #{index} device 2 (Allowed values: 0-4)	RO	Configuration dependent
[23:20]	num_dcs_p#{index}_d1	Number of device credited slices connected to port #{index} device 1 (Allowed values: 0-4)	RO	Configuration dependent
[19:16]	num_dcs_p#{index}_d0	Number of device credited slices connected to port #{index} device 0 (Allowed values: 0-4)	RO	Configuration dependent
[15:12]	Reserved	Reserved	RO	-
[11:8]	num_ccs_p#{index}	Number of CAL credited slices connected to port #{index} (Allowed values: 0-2)	RO	Configuration dependent
[7]	cal_connected_p#{index}	When set, CAL is connected on port #{index} (Allowed values: 0-1)	RO	Configuration dependent
[6:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:0]	device_type_p#{index}	Connected device type **5'b00000** Reserved **5'b00001** RN-I **5'b00010** RN-D **5'b00011** Reserved **5'b00100** RN-F_CHIB **5'b00101** RN-F_CHIB_ESAM **5'b00110** RN-F_CHIA **5'b00111** RN-F_CHIA_ESAM **5'b01000** HN-T **5'b01001** HN-I **5'b01010** HN-D **5'b01011** HN-P **5'b01100** SN-F_CHIC **5'b01101** SBSX **5'b01110** HN-F **5'b01111** SN-F_CHIE **5'b10000** SN-F_CHID **5'b10001** CXHA **5'b10010** CXRA **5'b10011** CXRH **5'b10100** RN-F_CHID **5'b10101** RN-F_CHID_ESAM **5'b10110** RN-F_CHIC **5'b10111** RN-F_CHIC_ESAM **5'b11000** RN-F_CHIE **5'b11001** RN-F_CHIE_ESAM **5'b11010** Reserved **5'b11011** Reserved **5'b11100** MTSX **5'b11101** HN-V **5'b11110** CCG **5'b11111** Reserved	RO	Configuration dependent

4.3.13.3 por_mxp_mesh_port_connect_info_east

Contains port connection information for East port.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h38`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-498: por_mxp_mesh_port_connect_info_east

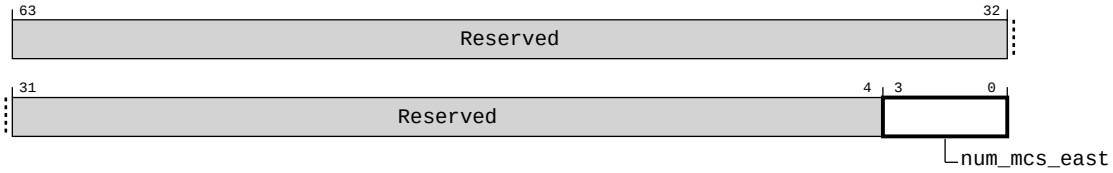


Table 4-514: por_mxp_mesh_port_connect_info_east attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	num_mcs_east	Number of mesh credited slices connected to East port (Allowed values: 0-4)	RO	Configuration dependent

4.3.13.4 por_mxp_mesh_port_connect_info_north

Contains port connection information for North port.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h40`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-499: por_mxp_mesh_port_connect_info_north

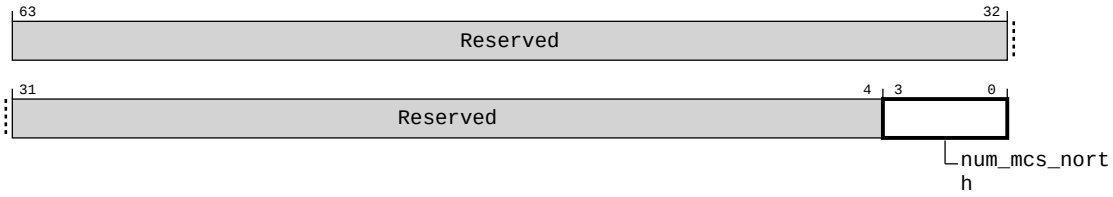


Table 4-515: por_mxp_mesh_port_connect_info_north attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	num_mcs_north	Number of mesh credited slices connected to North port (Allowed values: 0-4)	RO	Configuration dependent

4.3.13.5 por_mxp_device_port_connect_ldid_info_p0-5

There are 6 iterations of this register. The index ranges from 0 to 5. Contains LDID information for devices connected to port #{index}. Valid only for RNFs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h48 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-500: por_mxp_device_port_connect_ldid_info_p0-5

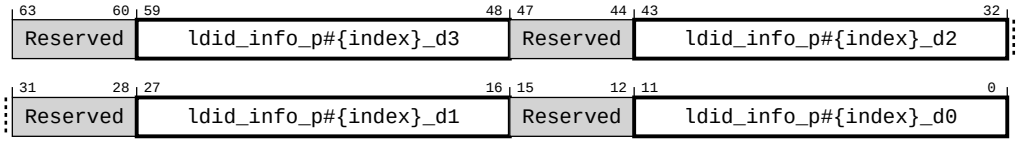


Table 4-516: por_mxp_device_port_connect_ldid_info_p0-5 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:48]	ldid_info_p#{index}_d3	LDID value of the device connected to port P#{index}_D3	RO	Configuration dependent
[47:44]	Reserved	Reserved	RO	-
[43:32]	ldid_info_p#{index}_d2	LDID value of the device connected to port P#{index}_D2	RO	Configuration dependent
[31:28]	Reserved	Reserved	RO	-
[27:16]	ldid_info_p#{index}_d1	LDID value of the device connected to port P#{index}_D1	RO	Configuration dependent
[15:12]	Reserved	Reserved	RO	-
[11:0]	ldid_info_p#{index}_d0	LDID value of the device connected to port P#{index}_D0	RO	Configuration dependent

4.3.13.6 por_mxp_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-501: por_mxp_child_info

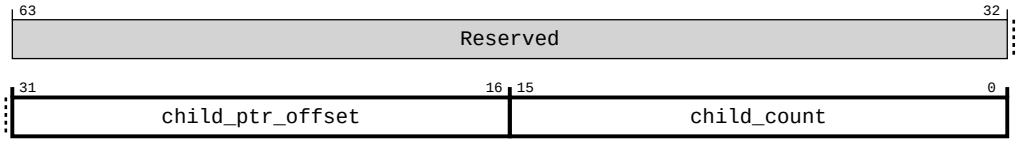


Table 4-517: por_mxp_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h100
[15:0]	child_count	Number of child nodes; used in discovery process	RO	Configuration dependent

4.3.13.7 por_mxp_child_pointer_0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Contains base address of the configuration subordinate for child #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h100 + #{8*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-502: por_mxp_child_pointer_0-31

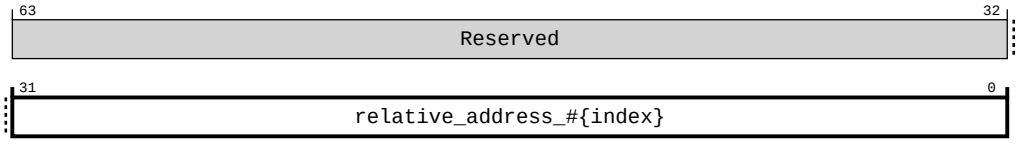


Table 4-518: por_mxp_child_pointer_0-31 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	relative_address_#{index}	Bit **[31]** External or internal child node **[30]** Set to 1'b0 Bits **[29:0]** Child node address offset relative to PERIPHBASE **1'b1** Indicates child pointer points to a configuration node that is external to CMN-700 **1'b0** Indicates child pointer points to a configuration node that is internal to CMN-700	RO	32'b0

4.3.13.8 por_mxp_p0-5_info

There are 6 iterations of this register. The index ranges from 0 to 5. Provides component identification information for XP port #{index}. NOTE: There will be max. of 6 MXP Port Info registers based on MXP_NUM_DEV_PORT_PARAM value. Each successive MXP Port Info register will be at the next 8 byte address boundary. Each successive MXP Port Info register will be named with the suffix. For example por_mxp_p<0:5>_info

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900 + #{16*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-503: por_mxp_p0-5_info

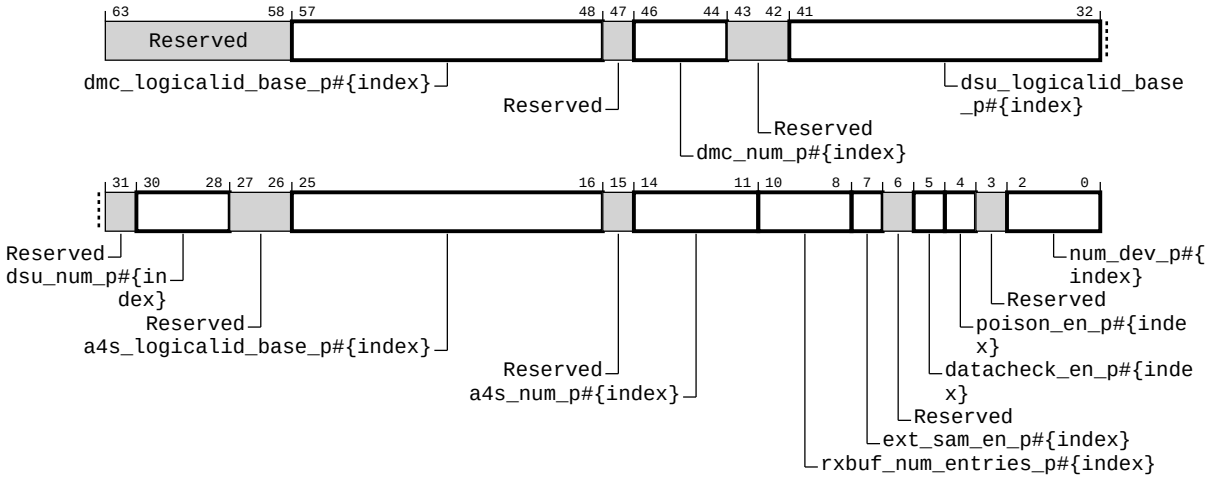


Table 4-519: por_mxp_p0-5_info attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	dmc_logicalid_base_p#{index}	DMC AXIU interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[47]	Reserved	Reserved	RO	-
[46:44]	dmc_num_p#{index}	Total number of SN-F AXIU interfaces at this port (0 to 4)	RO	Configuration dependent
[43:42]	Reserved	Reserved	RO	-
[41:32]	dsu_logicalid_base_p#{index}	DSU AXIU interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[31]	Reserved	Reserved	RO	-
[30:28]	dsu_num_p#{index}	Total number of RN-F AXIU interfaces at this port (0 to 4)	RO	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	a4s_logicalid_base_p#{index}	AXI4Stream interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[15]	Reserved	Reserved	RO	-
[14:11]	a4s_num_p#{index}	Total number of RN-F AXI4Stream interfaces at this port (0 to 4)	RO	Configuration dependent
[10:8]	rxbuf_num_entries_p#{index}	Number of input buffers at this port (2 to 4)	RO	Configuration dependent
[7]	ext_sam_en_p#{index}	ESAM enable	RO	Configuration dependent
[6]	Reserved	Reserved	RO	-
[5]	datacheck_en_p#{index}	Datacheck enable	RO	Configuration dependent
[4]	poison_en_p#{index}	Poison enable	RO	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2:0]	num_dev_p#{index}	Number of devices connected to this port (0 to 4)	RO	Configuration dependent

4.3.13.9 por_mxp_p0-5_info_1

There are 6 iterations of this register. The index ranges from 0 to 5. Provides component identification information for XP port #{index}. NOTE: There will be max. of 6 MXP Port Info registers based on MXP_NUM_DEV_PORT_PARAM value. Each successive MXP Port Info register will be at the next 8 byte address boundary. Each successive MXP Port Info register will be named with the suffix. For example por_mxp_p<0:5>_info_1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908 + #{16*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-504: por_mxp_p0-5_info_1

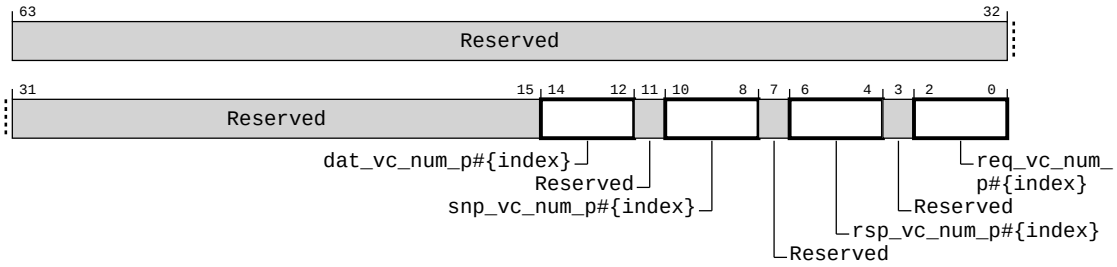


Table 4-520: por_mxp_p0-5_info_1 attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	dat_vc_num_p#{index}	Number of replicated channels on DAT VC at this port	RO	Configuration dependent
[11]	Reserved	Reserved	RO	-
[10:8]	snp_vc_num_p#{index}	Number of replicated channels on SNP VC at this port	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[7]	Reserved	Reserved	RO	-
[6:4]	rsp_vc_num_p#{index}	Number of replicated channels on RSP VC at this port	RO	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2:0]	req_vc_num_p#{index}	Number of replicated channels on REQ VC at this port	RO	Configuration dependent

4.3.13.10 por_dtm_unit_info

Provides component identification information for XP port 0 and 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h960`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-505: por_dtm_unit_info

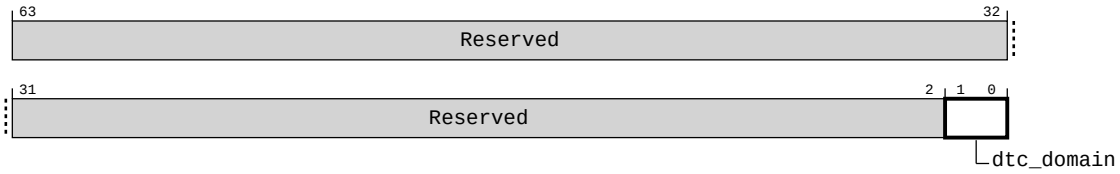


Table 4-521: por_dtm_unit_info attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	dtc_domain	DTC domain number associated with this DTM	RO	Configuration dependent

4.3.13.11 por_dtm_unit_info_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Provides component identification information for XP ports #{2*index} and #{(2*index)+1}. NOTE: There will be max. of 3 DTM Unit Info registers based on MXP_MULTIPLE_DTM_EN_PARAM and MXP_NUM_DEV_PORT_PARAM value. Each successive DTM Unit Info register will be at the next 8 byte address boundary. Each successive DTM Unit Info register will be named with the suffix corresponding to the DT register number. For example por_dtm_unit_info_dt<1:3>

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h968 + #{8*(index-1)}
```

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-506: por_dtm_unit_info_dt1-3

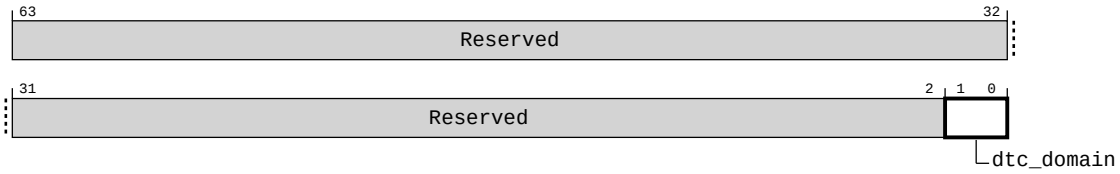


Table 4-522: por_dtm_unit_info_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	dtc_domain	DTC domain number associated with this DTM	RO	Configuration dependent

4.3.13.12 por_mxp_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-507: por_mxp_secure_register_groups_override

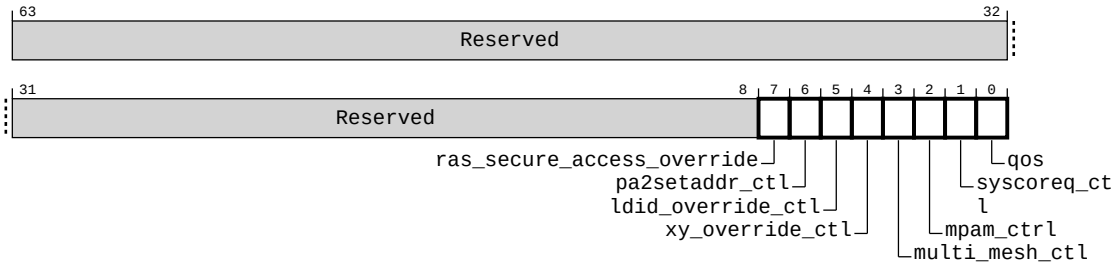


Table 4-523: por_mxp_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7]	ras_secure_access_override	Allow Non-secure access to Secure RAS registers	RW	1'b0
[6]	pa2setaddr_ctl	Allows Non-secure access to Secure PA to SETADDR control registers	RW	1'b0
[5]	ldid_override_ctl	Allows Non-secure access to Secure LDID override registers	RW	1'b0
[4]	xy_override_ctl	Allows Non-secure access to Secure XY override registers	RW	1'b0
[3]	multi_mesh_ctl	Allows Non-secure access to Secure Multi Mesh control registers	RW	1'b0
[2]	mpam_ctrl	Allows Non-secure access to Secure CHI port MPAM override register	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	syscoreq_ctl	Allows Non-secure access to Secure syscoreq_ctl registers	RW	1'b0
[0]	qos	Allows Non-secure access to Secure QoS registers	RW	1'b0

4.3.13.13 por_mxp_aux_ctl

Functions as the auxiliary control register for XP.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA00`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-508: por_mxp_aux_ctl

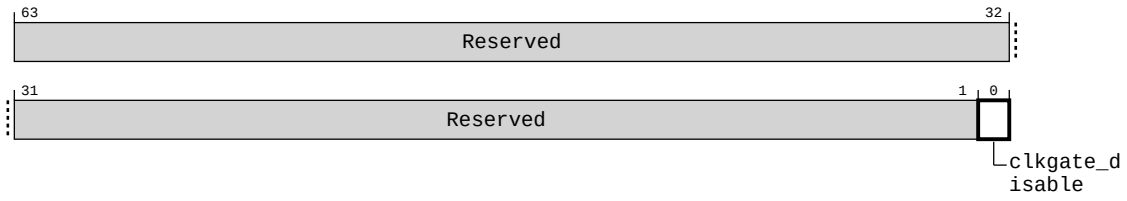


Table 4-524: por_mxp_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	clkgate_disable	Disables clock gating when set	RW	1'b0

4.3.13.14 por_mxp_device_port_ctl

Functions as the control register for XP device ports.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-509: por_mxp_device_port_ctl

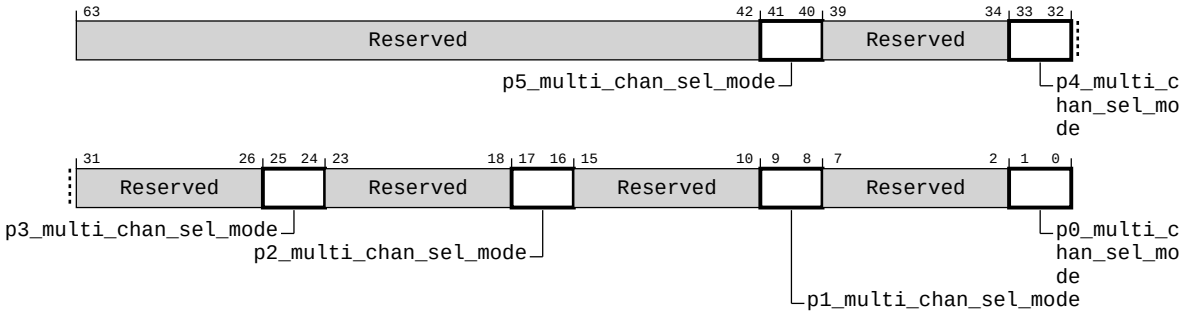


Table 4-525: por_mxp_device_port_ctl attributes

Bits	Name	Description	Type	Reset
[63:42]	Reserved	Reserved	RO	-
[41:40]	p5_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure **2'h0** Enable channel mapping based on TGTID scheme **2'h1** Enable channel mapping based on dynamic credit availability scheme **2'h2** Enable channel mapping based on direct connect scheme **2'h3** Reserved	RW	2'b0

Bits	Name	Description	Type	Reset
[39:34]	Reserved	Reserved	RO	-
[33:32]	p4_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure **2'h0** Enable channel mapping based on TGTID scheme **2'h1** Enable channel mapping based on dynamic credit availability scheme **2'h2** Enable channel mapping based on direct connect scheme **2'h3** Reserved	RW	2'b0
[31:26]	Reserved	Reserved	RO	-
[25:24]	p3_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure **2'h0** Enable channel mapping based on TGTID scheme **2'h1** Enable channel mapping based on dynamic credit availability scheme **2'h2** Enable channel mapping based on direct connect scheme **2'h3** Reserved	RW	2'b0
[23:18]	Reserved	Reserved	RO	-
[17:16]	p2_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure **2'h0** Enable channel mapping based on TGTID scheme **2'h1** Enable channel mapping based on dynamic credit availability scheme **2'h2** Enable channel mapping based on direct connect scheme **2'h3** Reserved	RW	2'b0
[15:10]	Reserved	Reserved	RO	-
[9:8]	p1_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure **2'h0** Enable channel mapping based on TGTID scheme **2'h1** Enable channel mapping based on dynamic credit availability scheme **2'h2** Enable channel mapping based on direct connect scheme **2'h3** Reserved	RW	2'b0
[7:2]	Reserved	Reserved	RO	-
[1:0]	p0_multi_chan_sel_mode	Selects the mode/scheme for channel selection in multi channel mesh structure **2'h0** Enable channel mapping based on TGTID scheme **2'h1** Enable channel mapping based on dynamic credit availability scheme **2'h2** Enable channel mapping based on direct connect scheme **2'h3** Reserved	RW	2'b0

4.3.13.15 por_mxp_p0-5_mpam_override

There are 6 iterations of this register. The index ranges from 0 to 5. Controls MPAM fields for devices connected to port #{index}. Valid only if the devices doesn't support MPAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA10 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.mpam_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-510: por_mxp_p0-5_mpam_override

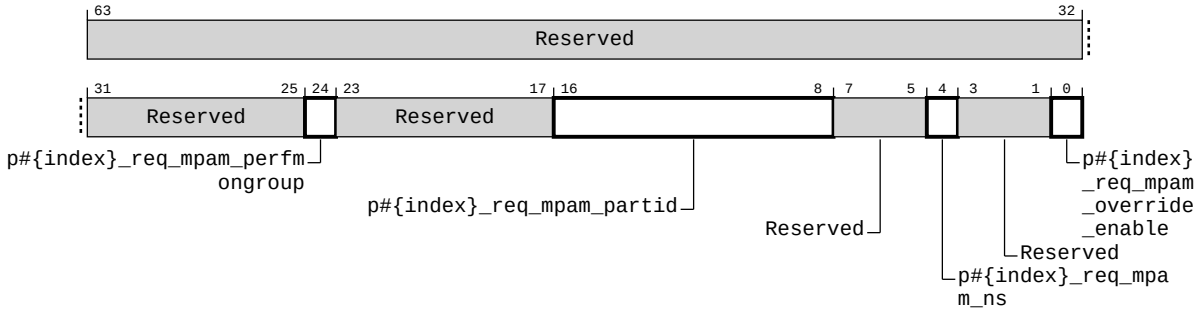


Table 4-526: por_mxp_p0-5_mpam_override attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	p#{index}_req_mpam_perfmongroup	MPAM.PerfMonGroup sub-field that overrides the REQ channel MPAM.PerfMonGroup when p#{index}_req_mpam_override_enable is set	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	p#{index}_req_mpam_partid	MPAM.PartID sub-field that overrides the REQ channel MPAM.PartID when p#{index}_req_mpam_override_enable is set	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	p#{index}_req_mpam_ns	MPAM.NS sub-field that overrides the REQ channel MPAM.NS when p#{index}_req_mpam_override_enable is set	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	p#{index}_req_mpam_override_enable	P#{index} DEV MPAM Override Enable on REQ Channel: 1 - Drive the MPAM fields on REQ channel with the values from this register, 0 - Override of MPAM fields in REQ channel is disabled	RW	1'b0

4.3.13.16 por_mxp_p0-5_ldid_override

There are 6 iterations of this register. The index ranges from 0 to 5. Controls LDID fields in REQ FLIT for devices connected to port #{index}. Valid only if POR_MXP_RNF_CLUSTER_EN_PARAM is 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA40 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.ldid_override_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-511: por_mxp_p0-5_ldid_override

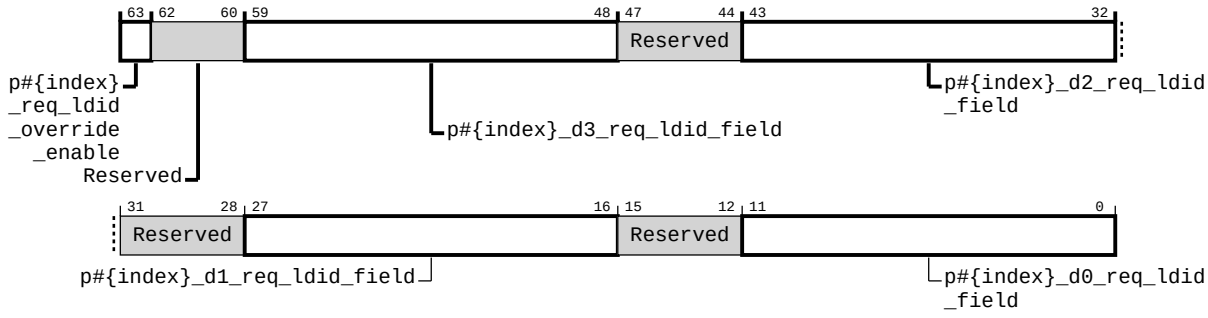


Table 4-527: por_mxp_p0-5_ldid_override attributes

Bits	Name	Description	Type	Reset
[63]	p#{index}_req_ldid_override_enable	P#{index} DEV LDID Override Enable on REQ Channel: 1 - Drive the LDID fields on REQ channel with the values from this register, 0 - Override of LDID fields in REQ channel is disabled	RW	1'b0
[62:60]	Reserved	Reserved	RO	-
[59:48]	p#{index}_d3_req_ldid_field	LDID value that overrides the P#{index}_D3 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	12'b0
[47:44]	Reserved	Reserved	RO	-
[43:32]	p#{index}_d2_req_ldid_field	LDID value that overrides the P#{index}_D2 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	12'b0
[31:28]	Reserved	Reserved	RO	-
[27:16]	p#{index}_d1_req_ldid_field	LDID value that overrides the P#{index}_D1 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	12'b0
[15:12]	Reserved	Reserved	RO	-
[11:0]	p#{index}_d0_req_ldid_field	LDID value that overrides the P#{index}_D0 REQ channel LDID field when p#{index}_req_ldid_override_enable is set	RW	12'b0

4.3.13.17 por_mxp_p0-5_qos_control

There are 6 iterations of this register. The index ranges from 0 to 5. Controls QoS settings for devices connected to port #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA80 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-512: por_mxp_p0-5_qos_control

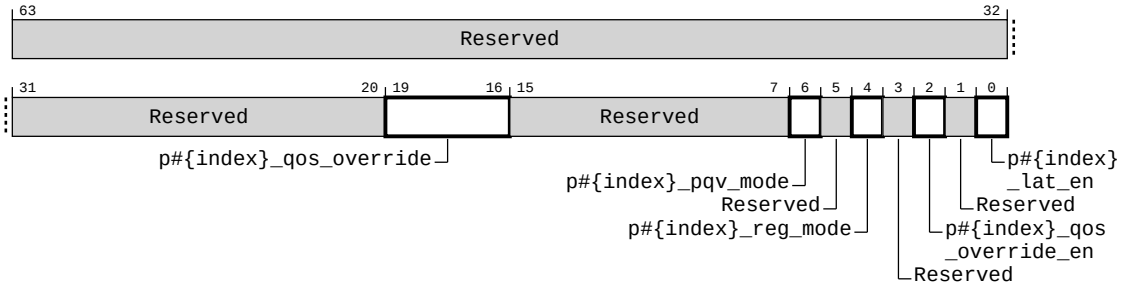


Table 4-528: por_mxp_p0-5_qos_control attributes

Bits	Name	Description	Type	Reset
[63:20]	Reserved	Reserved	RO	-
[19:16]	p#{index}_qos_override	QoS override value for port #{index}	RW	4'b0000
[15:7]	Reserved	Reserved	RO	-
[6]	p#{index}_pqv_mode	Configures the QoS regulator mode during period mode **1'b0** Normal mode; QoS value is stable when the manager is idle **1'b1** Quiesce high mode; QoS value tends to the maximum value when the manager is idle	RW	1'b0
[5]	Reserved	Reserved	RO	-
[4]	p#{index}_reg_mode	Configures the QoS regulator mode **1'b0** Latency mode **1'b1** Period mode; used for bandwidth regulation	RW	1'b0
[3]	Reserved	Reserved	RO	-
[2]	p#{index}_qos_override_en	Enables port #{index} QoS override; when set, allows QoS value on inbound transactions to be overridden	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	p#{index}_lat_en	Enables port #{index} QoS regulation when set	RW	1'b0

4.3.13.18 por_mxp_p0-5_qos_lat_tgt

There are 6 iterations of this register. The index ranges from 0 to 5. Controls QoS target latency/ period (in cycles) for regulation of devices connected to port #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA88 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-513: por_mxp_p0-5_qos_lat_tgt

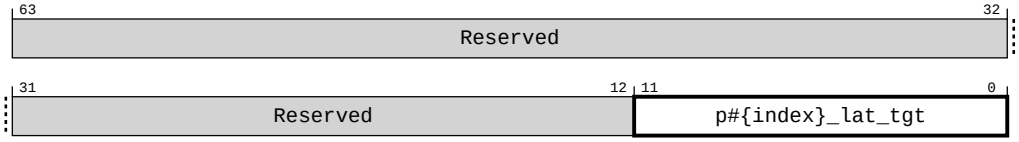


Table 4-529: por_mxp_p0-5_qos_lat_tgt attributes

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	-
[11:0]	p#{index}_lat_tgt	Port #{index} transaction target latency/period; a value of 0 corresponds to no regulation	RW	12'h000

4.3.13.19 por_mxp_p0-5_qos_lat_scale

There are 6 iterations of this register. The index ranges from 0 to 5. Controls the QoS target scale factor for devices connected to port #{index}. The scale factor is represented in powers of two from the range 2^(-3) to 2^(-10).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA90 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-514: por_mxp_p0-5_qos_lat_scale

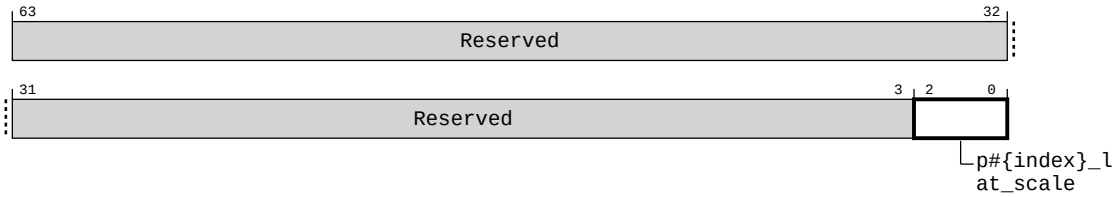


Table 4-530: por_mxp_p0-5_qos_lat_scale attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2:0]	p#{index}_lat_scale	Port 0 QoS scale factor **3'b000** 2^(-3) **3'b001** 2^(-4) **3'b010** 2^(-5) **3'b011** 2^(-6) **3'b100** 2^(-7) **3'b101** 2^(-8) **3'b110** 2^(-9) **3'b111** 2^(-10)	RW	3'h0

4.3.13.20 por_mxp_p0-5_qos_lat_range

There are 6 iterations of this register. The index ranges from 0 to 5. Controls the minimum and maximum QoS values generated by the QoS regulator for devices connected to port #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA98 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-515: por_mxp_p0-5_qos_lat_range

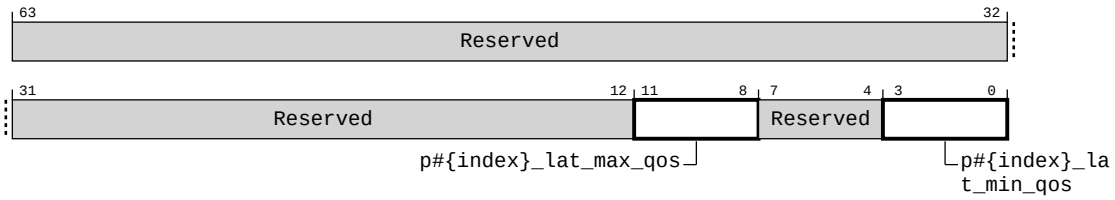


Table 4-531: por_mxp_p0-5_qos_lat_range attributes

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	-
[11:8]	p#{index}_lat_max_qos	Port #{index} QoS maximum value	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	p#{index}_lat_min_qos	Port #{index} QoS minimum value	RW	4'h0

4.3.13.21 por_mxp_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-516: por_mxp_pmu_event_sel

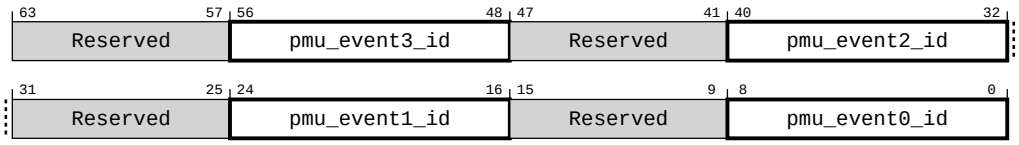


Table 4-532: por_mxp_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	pmu_event3_id	XP PMU Event 3 ID; see pmu_event0_id for encodings	RW	9'b0
[47:41]	Reserved	Reserved	RO	-
[40:32]	pmu_event2_id	XP PMU Event 2 ID; see pmu_event0_id for encodings	RW	9'b0
[31:25]	Reserved	Reserved	RO	-
[24:16]	pmu_event1_id	XP PMU Event 1 ID; see pmu_event0_id for encodings	RW	9'b0
[15:9]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[8:0]	pmu_event0_id	XP PMU Event 0 ID Bits [8:5]: **4'b0000** REQ; REQ channel when POR_REQ_VC_NUM_PARAM = 1 ; REQ Sub-channel 1: when POR_REQ_VC_NUM_PARAM > 1 **4'b0001** RSP; RSP channel when POR_RSP_VC_NUM_PARAM = 1 ; RSP Sub-channel 1: when POR_RSP_VC_NUM_PARAM > 1 **4'b0010** SNP; SNP channel when POR_SNP_VC_NUM_PARAM = 1 ; SNP Sub-channel 1: when POR_SNP_VC_NUM_PARAM > 1 **4'b0011** DAT; DAT channel when POR_DAT_VC_NUM_PARAM = 1 ; DAT Sub-channel 1: when POR_DAT_VC_NUM_PARAM > 1 **4'b0100** PUB **4'b0101** RSP2; RSP Sub-channel 2: Applicable when POR_RSP_VC_NUM_PARAM > 1 **4'b0110** DAT2; DAT Sub-channel 2: Applicable when POR_DAT_VC_NUM_PARAM > 1 **4'b0111** REQ2; REQ Sub-channel 2: Applicable when POR_REQ_VC_NUM_PARAM > 1 **4'b1000** SNP2; SNP Sub-channel 2: Applicable when POR_SNP_VC_NUM_PARAM > 1 **4'b1100** AXI W Subordinate Error **4'b1101** AXI W Decode Error **4'b1110** PA out of range Error **3'b000** East when NUM_XP > 1 ; Device port 0 when NUM_XP == 1 (Single XP config) **3'b001** West when NUM_XP > 1 ; Device port 1 when NUM_XP == 1 (Single XP config) **3'b010** North when NUM_XP > 1 ; Device port 2 when NUM_XP == 1 (Single XP config) **3'b011** South when NUM_XP > 1 ; Device port 3 when NUM_XP == 1 (Single XP config) **3'b100** Device port 0 when NUM_XP > 1 ; Device port 4 when NUM_XP == 1 (Single XP config) **3'b101** Device port 1 when NUM_XP > 1 ; Device port 5 when NUM_XP == 1 (Single XP config) **3'b110** Device port 2 when NUM_XP > 1 ; No Selection when NUM_XP == 1 (Single XP config) **3'b111** Device port 3 when NUM_XP > 1 ; No Selection when NUM_XP == 1 (Single XP config) Bits [1:0]: Event specifier **2'b00** No event **2'b01** TX flit valid; signaled when a flit is successfully transmitted **2'b10** TX flit stall; signaled when flit transmission is stalled and waiting on credits **2'b11** Partial DAT flit; signaled when 128-bit DAT flits could not be merged into a 256-bit DAT flit; only applicable on the DAT PC on RN-F CHIA and RN-F CHIA ESAM ports	RW	9'b0

4.3.13.22 por_mxp_errfr

Functions as the error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3000`

Type

RO

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-517: por_mxp_errfr

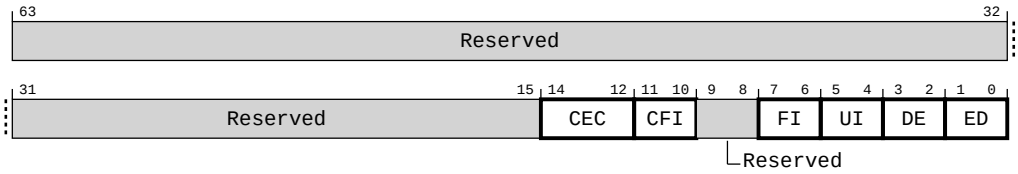


Table 4-533: por_mxp_errfr attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism 3'b000: Does not implement standardized error counter model	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors for data poison	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.13.23 por_mxp_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3008`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-518: por_mxp_errctlr

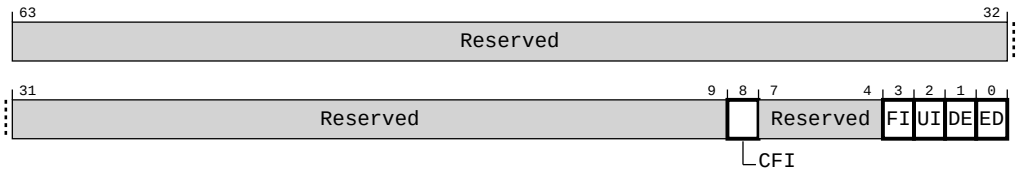


Table 4-534: por_mxp_errctlr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_mxp_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_mxp_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_mxp_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_mxp_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_mxp_errfr.ED	RW	1'b0

4.3.13.24 por_mxp_errstatus

Functions as the error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3010`

Type

W1C

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-519: por_mxp_errstatus

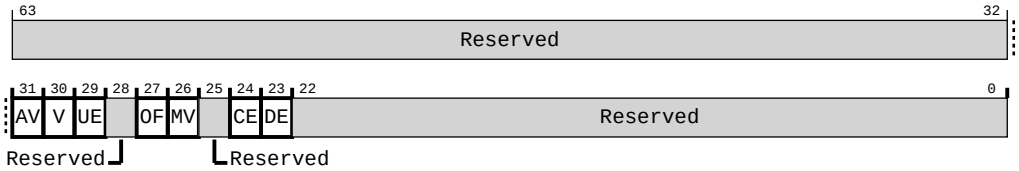


Table 4-535: por_mxp_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid **1'b0** Address is not valid	W1C	1'b0

Bits	Name	Description	Type	Reset
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	por_mxp_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.13.25 por_mxp_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3028`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-520: por_mxp_errmisc

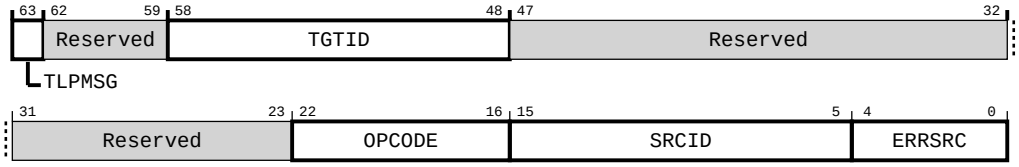


Table 4-536: por_mxp_errmisc attributes

Bits	Name	Description	Type	Reset
[63]	TLPMSG	Error flit TLPMSG Status	RW	1'b0
[62:59]	Reserved	Reserved	RO	-
[58:48]	TGTID	Error flit target ID	RW	11'b0
[47:23]	Reserved	Reserved	RO	-
[22:16]	OPCODE	Error flit opcode	RW	7'b0
[15:5]	SRCID	Error flit source ID	RW	11'b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	Error source for Mesh With Replicated Channels: Bits [4:2]: Transaction type **3'b000** REQ **3'b001** RSP **3'b010** SNP **3'b011** DAT **3'b100** REQ2 **3'b101** RSP2 **3'b110** SNP2 **3'b111** DAT2 Bits [1:0]: Port **2'b00** Port 0 **2'b01** Port 1 **2'b10** Port 2 **2'b11** Port 3	RW	5'b0
[4:0]	ERRSRC	Mesh Without Replicated Channels: Bits [4:2]: Transaction type **3'b000** REQ **3'b001** RSP **3'b010** SNP **3'b011** DAT **3'b100** Reserved **3'b101** Reserved **3'b110** Reserved **3'b111** Reserved Bits [1:0]: Port **2'b00** Port 0 **2'b01** Port 1 **2'b10** Port 2 **2'b11** Port 3	RW	5'b0

4.3.13.26 por_mxp_p0-5_byte_par_err_inj

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the byte parity error injection register for XP port #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h3030 + #{8*index}
```

Type

WO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-521: por_mxp_p0-5_byte_par_err_inj

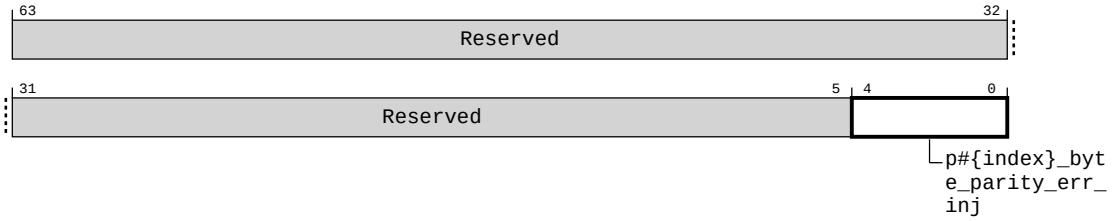


Table 4-537: por_mxp_p0-5_byte_par_err_inj attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:0]	p#{index}_byte_parity_err_inj	Specifies a byte lane; once this register is written, a byte parity error is injected in the specified byte lane on the next DAT flit upload NOTE: Only applicable if an RN-F is attached to port #{index}. Byte parity error is only injected if the RN-F is configured to not support Datacheck.	WO	5'h00

4.3.13.27 por_mxp_errfr_NS

Functions as the Non-secure error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3100`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-522: por_mxp_errfr_NS

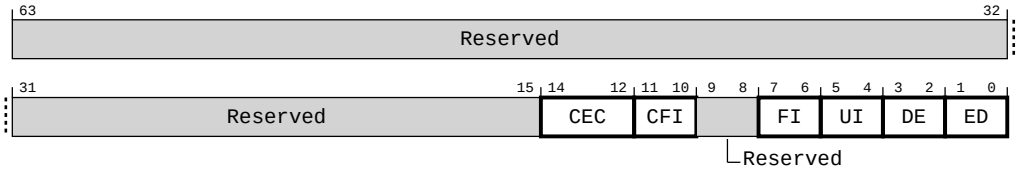


Table 4-538: por_mxp_errfr_NS attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism 3'b000: Does not implement standardized error counter model	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors for data poison	RO	2'b01
[1:0]	ED	Error detection	RO	2'b01

4.3.13.28 por_mxp_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3108

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-523: por_mxp_errctlr_NS

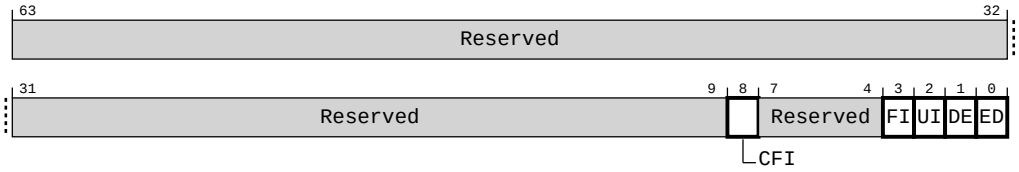


Table 4-539: por_mxp_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_mxp_errfr_NS.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_mxp_errfr_NS.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_mxp_errfr_NS.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_mxp_errfr_NS.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_mxp_errfr_NS.ED	RW	1'b0

4.3.13.29 por_mxp_errstatus_NS

Functions as the Non-secure error status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3110`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-524: por_mxp_errstatus_NS

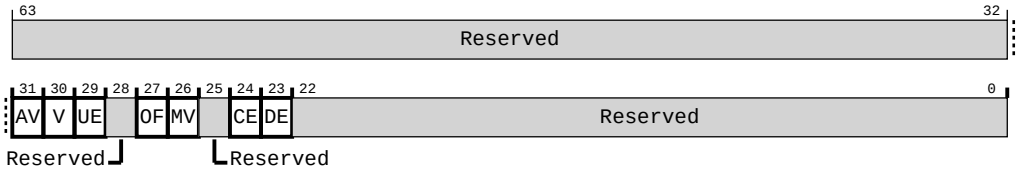


Table 4-540: por_mxp_errstatus_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0

Bits	Name	Description	Type	Reset
[26]	MV	por_mxp_errmisc_NS valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.13.30 por_mxp_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3128`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-525: por_mxp_errmisc_NS

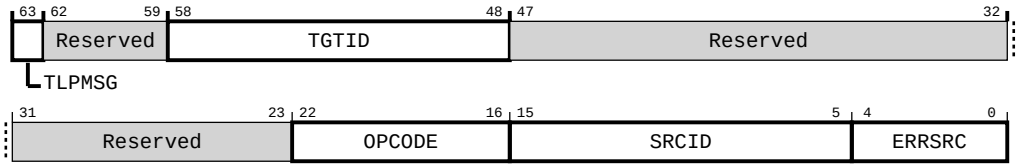


Table 4-541: por_mxp_errmisc_NS attributes

Bits	Name	Description	Type	Reset
[63]	TLPMSG	Error flit TLPMSG Status	RW	1'b0
[62:59]	Reserved	Reserved	RO	-
[58:48]	TGTID	Error flit target ID	RW	11'b0
[47:23]	Reserved	Reserved	RO	-
[22:16]	OPCODE	Error flit opcode	RW	7'b0
[15:5]	SRCID	Error flit source ID	RW	11'b0
[4:0]	ERRSRC	Error source for Mesh With Replicated Channels: Bits [4:2]: Transaction type **3'b000** REQ **3'b001** RSP **3'b010** SNP **3'b011** DAT **3'b100** REQ2 **3'b101** RSP2 **3'b110** SNP2 **3'b111** DAT2 Bits [1:0]: Port **2'b00** Port 0 **2'b01** Port 1 **2'b10** Port 2 **2'b11** Port 3	RW	5'b0

Bits	Name	Description	Type	Reset
[4:0]	ERRSRC	Mesh Without Replicated Channels: Bits [4:2]: Transaction type **3'b000** REQ **3'b001** RSP **3'b010** SNP **3'b011** DAT **3'b100** Reserved **3'b101** Reserved **3'b110** Reserved **3'b111** Reserved Bits [1:0]: Port **2'b00** Port 0 **2'b01** Port 1 **2'b10** Port 2 **2'b11** Port 3	RW	5'b0

4.3.13.31 por_mxp_p0-5_syscoreq_ctl

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the port #{index} snoop and DVM domain control register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por_mxp_p#{index}_syscoack_status. NOTE: Only valid on RN-F ports.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C00 + #{16*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.syscoreq_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-526: por_mxp_p0-5_syscoreq_ctl

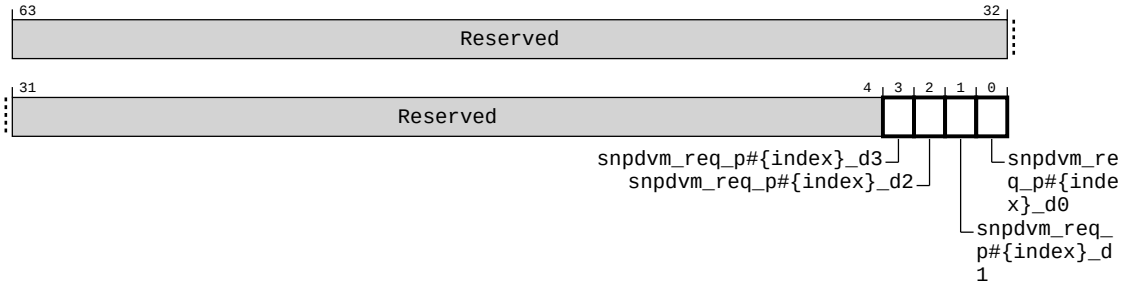


Table 4-542: por_mxp_p0-5_syscoreq_ctl attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	snpdvm_req_p#{index}_d3	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 3 on port #{index}	RW	1'b0
[2]	snpdvm_req_p#{index}_d2	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 2 on port #{index}	RW	1'b0
[1]	snpdvm_req_p#{index}_d1	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 1 on port #{index}	RW	1'b0
[0]	snpdvm_req_p#{index}_d0	When set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 0 on port #{index}	RW	1'b0

4.3.13.32 por_mxp_p0-5_syscoack_status

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the port #{index} snoop and DVM domain status register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por_mxp_p#{index}_syscoreq_ctl. NOTE: Only valid on RN-F ports.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1C08 + #{16*index}`

Type

RO

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.syscoreq_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-527: por_mxp_p0-5_syscoack_status

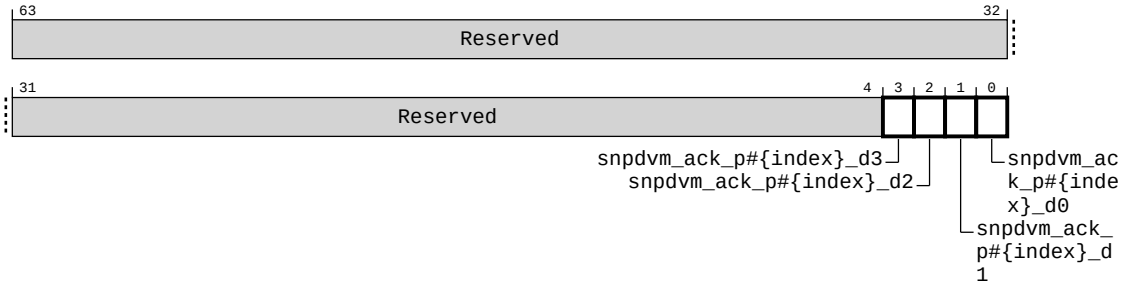


Table 4-543: por_mxp_p0-5_syscoack_status attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	snpdvm_ack_p#{index}_d3	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 3 on port #{index}	RO	1'b0
[2]	snpdvm_ack_p#{index}_d2	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 2 on port #{index}	RO	1'b0
[1]	snpdvm_ack_p#{index}_d1	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 1 on port #{index}	RO	1'b0
[0]	snpdvm_ack_p#{index}_d0	When set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 0 on port #{index}	RO	1'b0

4.3.13.33 por_dtm_control

Functions as the DTM control register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2100`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-528: por_dtm_control

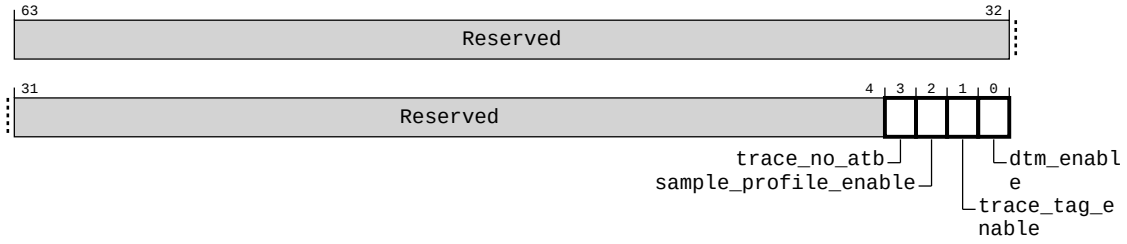


Table 4-544: por_dtm_control attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	trace_no_atb	When set, trace packet is not delivered out of ATB, and FIFO entry holds the first trace packet. NOTE: if any MXP has this bit set, ATB protocol will not be functional.	RW	1'b0
[2]	sample_profile_enable	Enables sample profile function	RW	1'b0
[1]	trace_tag_enable	Watchpoint trace tag enable 1'b1: Trace tag enabled 1'b0: No trace tag	RW	1'b0
[0]	dtm_enable	Enables debug watchpoint and PMU function; prior to writing this bit, all other DT configuration registers must be programmed; once this bit is set, other DT configuration registers must not be modified	RW	1'b0

4.3.13.34 por_dtm_fifo_entry_ready

Controls status of DTM FIFO entries.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2118`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-529: por_dtm_fifo_entry_ready

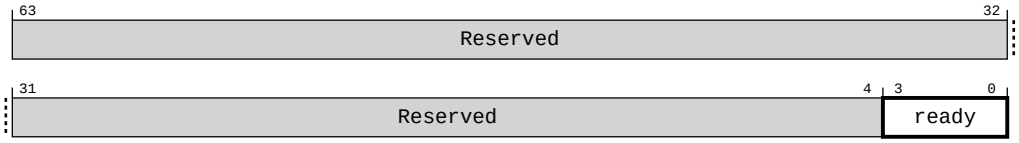


Table 4-545: por_dtm_fifo_entry_ready attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	ready	Indicates which DTM FIFO entries are ready; write a 1 to clear Bit [3]: Entry 3 ready when set Bit [2]: Entry 2 ready when set Bit [1]: Entry 1 ready when set Bit [0]: Entry 0 ready when set	W1C	4'b0

4.3.13.35 por_dtm_fifo_entry0-3_0

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{index} data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2120 + #{24*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-530: por_dtm_fifo_entry0-3_0

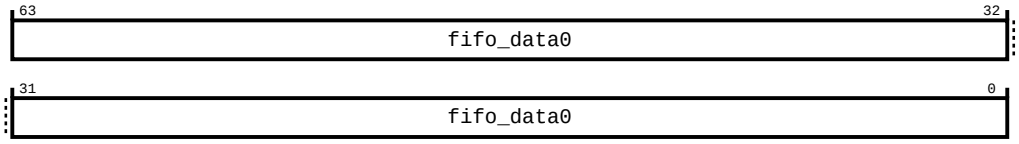


Table 4-546: por_dtm_fifo_entry0-3_0 attributes

Bits	Name	Description	Type	Reset
[63:0]	fifo_data0	Entry data bit vector 63:0	RO	64'b0

4.3.13.36 por_dtm_fifo_entry0-3_1

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{index} data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2128 + #{24*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-531: por_dtm_fifo_entry0-3_1

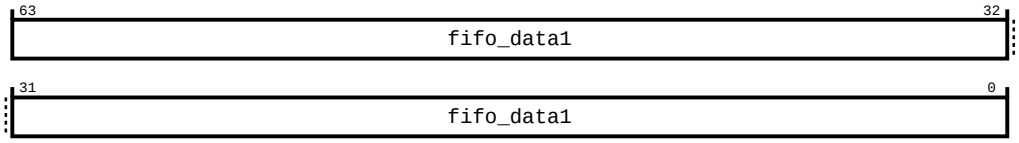


Table 4-547: por_dtm_fifo_entry0-3_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	fifo_data1	Entry data bit vector 127:64	RO	64'b0

4.3.13.37 por_dtm_fifo_entry0-3_2

There are 4 iterations of this register. The index ranges from 0 to 3. Contains DTM FIFO entry #{index} data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2130 + #{24*index}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-532: por_dtm_fifo_entry0-3_2

Table 4-548: por_dtm_fifo_entry0-3_2 attributes

Bits	Name	Description	Type	Reset
[63:48]	fifo_cycle_count	Entry cycle count bit vector 15:0	RO	16'b0
[47:0]	fifo_data2	Entry data bit vector 143:128	RO	48'b0

4.3.13.38 por_dtm_wp0-3_config

There are 4 iterations of this register. The index ranges from 0 to 3. Configures watchpoint #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h21A0 + #{24*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-533: por_dtm_wp0-3_config

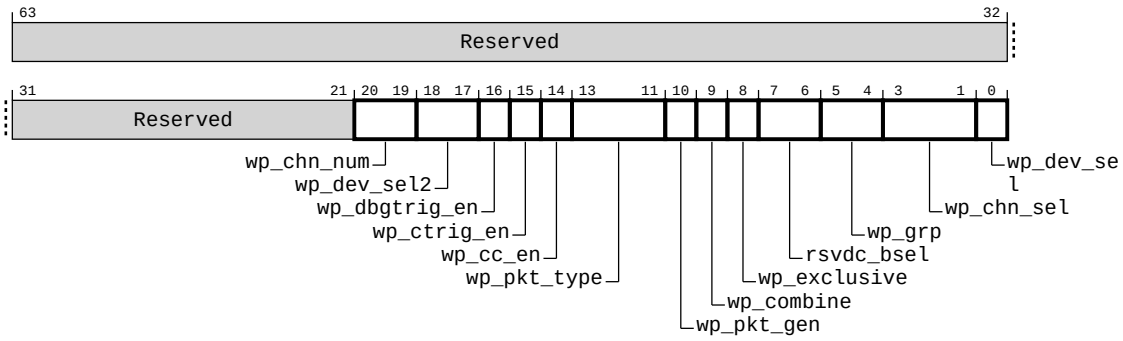


Table 4-549: por_dtm_wp0-3_config attributes

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[20:19]	wp_chn_num	VC number index for replicated channels in specified SMXP	RW	2'b0
[18:17]	wp_dev_sel2	Upper bits for device port selection in specified SMXP	RW	2'b0
[16]	wp_dbgtrig_en	Enables watchpoint debug trigger packet generation	RW	1'b0
[15]	wp_ctrig_en	Enables watchpoint cross trigger packet generation	RW	1'b0
[14]	wp_cc_en	Enables inclusion of cycle count in watchpoint track packet generation	RW	1'b0
[13:11]	wp_pkt_type	Trace packet type **3'b000** TXNID (up to X18) **3'b001** TXNID + opcode (up to X9) **3'b010** TXNID + opcode + source ID + target ID (up to X4) **3'b011** Reserved **3'b100** Control flit **3'b101** DAT flit DATA [127:0] **3'b110** DAT flit DATA [255:128] **3'b111** Reserved	RW	3'b000
[10]	wp_pkt_gen	Enables watchpoint trace packet generation	RW	1'b0
[9]	wp_combine	Enables combination of watchpoints #{index} and #{index+1}	RW	1'b0
[8]	wp_exclusive	Watchpoint mode **1'b0** Regular mode **1'b1** Exclusive mode	RW	1'b0
[7:6]	rsvdc_bsel	Byte select of RSVDC in trace packet **2'h0** Select RSVDC[7:0] **2'h1** Select RSVDC[15:8] **2'h2** Select RSVDC[23:16] **2'h3** Select RSVDC[31:24]	RW	1'b0
[5:4]	wp_grp	Watchpoint register format group **2'h0** Select primary group **2'h1** Select secondary group **2'h2** Select tertiary group **2'h3** Reserved	RW	1'b0
[3:1]	wp_chn_sel	VC selection **3'b000** Select REQ VC **3'b001** Select RSP VC **3'b010** Select SNP VC **3'b011** Select DATA VC All other values are reserved.	RW	3'b000
[0]	wp_dev_sel	Device port selection in specified SMXP **1'b0** Select device port 0 **1'b1** Select device port 1	RW	1'b0

4.3.13.39 por_dtm_wp0-3_val

There are 4 iterations of this register. The index ranges from 0 to 3. Configures watchpoint #{index} comparison value.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h21A8 + #{24*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-534: por_dtm_wp0-3_val

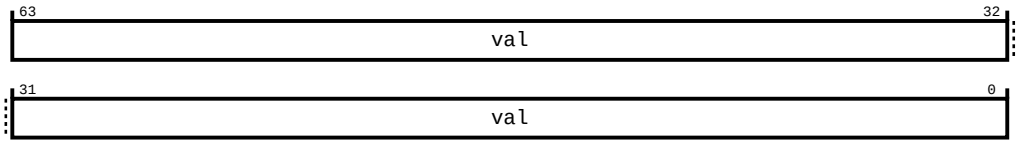


Table 4-550: por_dtm_wp0-3_val attributes

Bits	Name	Description	Type	Reset
[63:0]	val	Refer to DTM watchpoint section for details	RW	64'b0

4.3.13.40 por_dtm_wp0-3_mask

There are 4 iterations of this register. The index ranges from 0 to 3. Configures watchpoint #{index} comparison mask.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h21B0 + #{24*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-535: por_dtm_wp0-3_mask

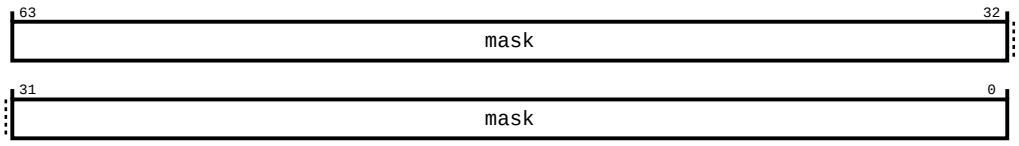


Table 4-551: por_dtm_wp0-3_mask attributes

Bits	Name	Description	Type	Reset
[63:0]	mask	Refer to DTM watchpoint section for details	RW	64'b0

4.3.13.41 por_dtm_pmsicr

Functions as the sampling interval counter register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2200`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-536: por_dtm_pmsicr

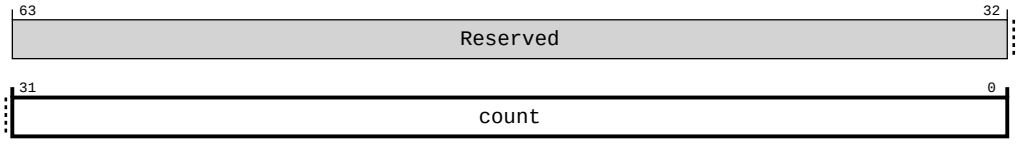


Table 4-552: por_dtm_pmsicr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	count	Current value of sample counter	RW	32'b0

4.3.13.42 por_dtm_pmsirr

Functions as the sampling interval reload register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2208`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-537: por_dtm_pmsirr

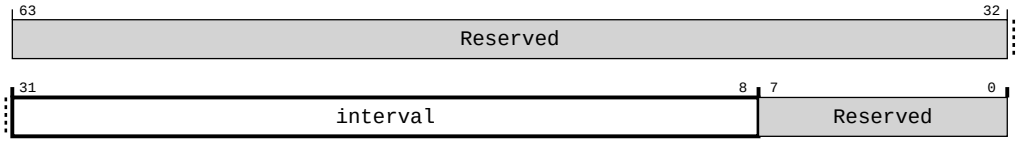


Table 4-553: por_dtm_pmsirr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:8]	interval	Sampling interval to be reloaded	RW	24'b0
[7:0]	Reserved	Reserved	RO	-

4.3.13.43 por_dtm_pmu_config

Configures the DTM PMU.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2210`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-538: por_dtm_pmu_config

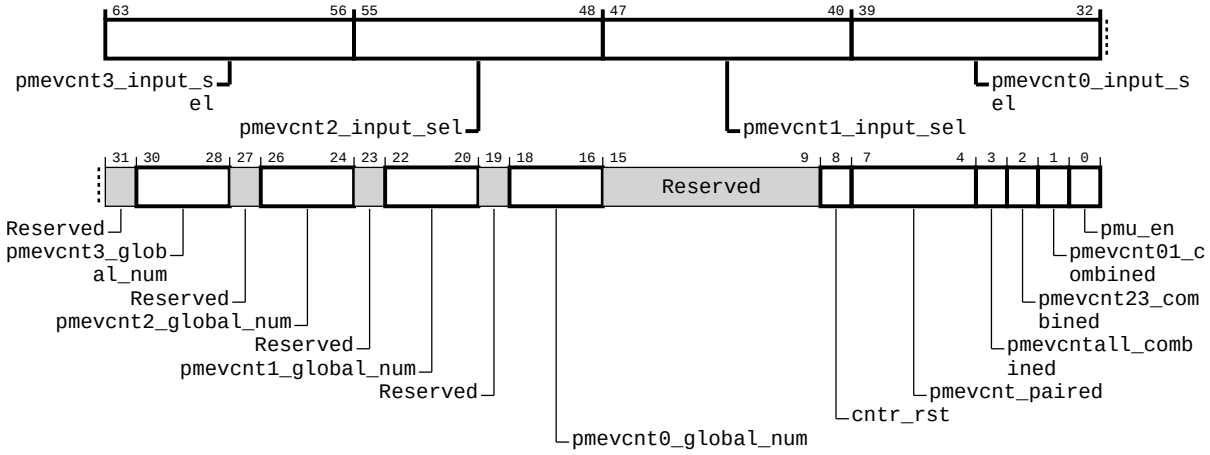


Table 4-554: por_dtm_pmu_config attributes

Bits	Name	Description	Type	Reset
[63:56]	pmevcnt3_input_sel	Source to be counted in PMU counter 3; see pmevcnt0_input_sel for encodings	RW	8'b0
[55:48]	pmevcnt2_input_sel	Source to be counted in PMU counter 2; see pmevcnt0_input_sel for encodings	RW	8'b0
[47:40]	pmevcnt1_input_sel	Source to be counted in PMU counter 1; see pmevcnt0_input_sel for encodings	RW	8'b0
[39:32]	pmevcnt0_input_sel	Source to be counted in PMU counter 0: Port2, Port3, Port4 and Port5 encodings are applicable when (MXP_NUM_DEV_PORT_PARAM > 2 and MXP_MULTIPLE_DTM_EN_PARAM = 0) **8'h00** Watchpoint 0 **8'h01** Watchpoint 1 **8'h02** Watchpoint 2 **8'h03** Watchpoint 3 **8'h04** XP PMU Event 0 **8'h05** XP PMU Event 1 **8'h06** XP PMU Event 2 **8'h07** XP PMU Event 3	RW	8'b0
[39:32]	pmevcnt0_input_sel	**8'h10** Port 0 Device 0 PMU Event 0 **8'h11** Port 0 Device 0 PMU Event 1 **8'h12** Port 0 Device 0 PMU Event 2 **8'h13** Port 0 Device 0 PMU Event 3 **8'h14** Port 0 Device 1 PMU Event 0 **8'h15** Port 0 Device 1 PMU Event 1 **8'h16** Port 0 Device 1 PMU Event 2 **8'h17** Port 0 Device 1 PMU Event 3 **8'h18** Port 0 Device 2 PMU Event 0 **8'h19** Port 0 Device 2 PMU Event 1 **8'h1A** Port 0 Device 2 PMU Event 2 **8'h1B** Port 0 Device 2 PMU Event 3 **8'h1C** Port 0 Device 3 PMU Event 0 **8'h1D** Port 0 Device 3 PMU Event 1 **8'h1E** Port 0 Device 3 PMU Event 2 **8'h1F** Port 0 Device 3 PMU Event 3	RW	8'b0

Bits	Name	Description		Type	Reset
[39:32]	pmevcnt0_input_sel	**8'h20**	Port 1 Device 0 PMU Event 0	RW	8'b0
		8'h21	Port 1 Device 0 PMU Event 1		
		8'h22	Port 1 Device 0 PMU Event 2		
		8'h23	Port 1 Device 0 PMU Event 3		
		8'h24	Port 1 Device 1 PMU Event 0		
		8'h25	Port 1 Device 1 PMU Event 1		
		8'h26	Port 1 Device 1 PMU Event 2		
		8'h27	Port 1 Device 1 PMU Event 3		
		8'h28	Port 1 Device 2 PMU Event 0		
		8'h29	Port 1 Device 2 PMU Event 1		
		8'h2A	Port 1 Device 2 PMU Event 2		
		8'h2B	Port 1 Device 2 PMU Event 3		
		8'h2C	Port 1 Device 3 PMU Event 0		
		8'h2D	Port 1 Device 3 PMU Event 1		
		8'h2E	Port 1 Device 3 PMU Event 2		
		8'h2F	Port 1 Device 3 PMU Event 3		
[39:32]	pmevcnt0_input_sel	**8'h30**	Port 2 Device 0 PMU Event 0	RW	8'b0
		8'h31	Port 2 Device 0 PMU Event 1		
		8'h32	Port 2 Device 0 PMU Event 2		
		8'h33	Port 2 Device 0 PMU Event 3		
		8'h34	Port 2 Device 1 PMU Event 0		
		8'h35	Port 2 Device 1 PMU Event 1		
		8'h36	Port 2 Device 1 PMU Event 2		
		8'h37	Port 2 Device 1 PMU Event 3		
		8'h38	Port 2 Device 2 PMU Event 0		
		8'h39	Port 2 Device 2 PMU Event 1		
		8'h3A	Port 2 Device 2 PMU Event 2		
		8'h3B	Port 2 Device 2 PMU Event 3		
		8'h3C	Port 2 Device 3 PMU Event 0		
		8'h3D	Port 2 Device 3 PMU Event 1		
		8'h3E	Port 2 Device 3 PMU Event 2		
		8'h3F	Port 2 Device 3 PMU Event 3		
[39:32]	pmevcnt0_input_sel	**8'h40**	Port 3 Device 0 PMU Event 0	RW	8'b0
		8'h41	Port 3 Device 0 PMU Event 1		
		8'h42	Port 3 Device 0 PMU Event 2		
		8'h43	Port 3 Device 0 PMU Event 3		
		8'h44	Port 3 Device 1 PMU Event 0		
		8'h45	Port 3 Device 1 PMU Event 1		
		8'h46	Port 3 Device 1 PMU Event 2		
		8'h47	Port 3 Device 1 PMU Event 3		
		8'h48	Port 3 Device 2 PMU Event 0		
		8'h49	Port 3 Device 2 PMU Event 1		
		8'h4A	Port 3 Device 2 PMU Event 2		
		8'h4B	Port 3 Device 2 PMU Event 3		
		8'h4C	Port 3 Device 3 PMU Event 0		
		8'h4D	Port 3 Device 3 PMU Event 1		
		8'h4E	Port 3 Device 3 PMU Event 2		
		8'h4F	Port 3 Device 3 PMU Event 3		

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	**8'h50** Port 4 Device 0 PMU Event 0 **8'h51** Port 4 Device 0 PMU Event 1 **8'h52** Port 4 Device 0 PMU Event 2 **8'h53** Port 4 Device 0 PMU Event 3 **8'h54** Port 4 Device 1 PMU Event 0 **8'h55** Port 4 Device 1 PMU Event 1 **8'h56** Port 4 Device 1 PMU Event 2 **8'h57** Port 4 Device 1 PMU Event 3 **8'h58** Port 4 Device 2 PMU Event 0 **8'h59** Port 4 Device 2 PMU Event 1 **8'h5A** Port 4 Device 2 PMU Event 2 **8'h5B** Port 4 Device 2 PMU Event 3 **8'h5C** Port 4 Device 3 PMU Event 0 **8'h5D** Port 4 Device 3 PMU Event 1 **8'h5E** Port 4 Device 3 PMU Event 2 **8'h5F** Port 4 Device 3 PMU Event 3	RW	8'b0
[39:32]	pmevcnt0_input_sel	**8'h60** Port 5 Device 0 PMU Event 0 **8'h61** Port 5 Device 0 PMU Event 1 **8'h62** Port 5 Device 0 PMU Event 2 **8'h63** Port 5 Device 0 PMU Event 3 **8'h64** Port 5 Device 1 PMU Event 0 **8'h65** Port 5 Device 1 PMU Event 1 **8'h66** Port 5 Device 1 PMU Event 2 **8'h67** Port 5 Device 1 PMU Event 3 **8'h68** Port 5 Device 2 PMU Event 0 **8'h69** Port 5 Device 2 PMU Event 1 **8'h6A** Port 5 Device 2 PMU Event 2 **8'h6B** Port 5 Device 2 PMU Event 3 **8'h6C** Port 5 Device 3 PMU Event 0 **8'h6D** Port 5 Device 3 PMU Event 1 **8'h6E** Port 5 Device 3 PMU Event 2 **8'h6F** Port 5 Device 3 PMU Event 3	RW	8'b0
[31]	Reserved	Reserved	RO	-
[30:28]	pmevcnt3_global_num	Global counter to pair with PMU counter 3; see pmevcnt0_global_num for encodings	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	pmevcnt2_global_num	Global counter to pair with PMU counter 2; see pmevcnt0_global_num for encodings	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	pmevcnt1_global_num	Global counter to pair with PMU counter 1; see pmevcnt0_global_num for encodings	RW	3'b0
[19]	Reserved	Reserved	RO	-
[18:16]	pmevcnt0_global_num	Global counter to pair with PMU counter 0 **3'b000** Global PMU event counter A **3'b001** Global PMU event counter B **3'b010** Global PMU event counter C **3'b011** Global PMU event counter D **3'b100** Global PMU event counter E **3'b101** Global PMU event counter F **3'b110** Global PMU event counter G **3'b111** Global PMU event counter H	RW	3'b0
[15:9]	Reserved	Reserved	RO	-
[8]	cntr_rst	Enables clearing of live counters upon assertion of snapshot	RW	1'b0

Bits	Name	Description	Type	Reset
[7:4]	pmevcnt_paired	PMU local counter paired with global counter	RW	4'b0
[3]	pmevcntall_combined	Enables combination of all PMU counters (0, 1, 2, 3) NOTE: When set, pmevcnt01_combined and pmevcnt23_combined have no effect.	RW	1'b0
[2]	pmevcnt23_combined	Enables combination of PMU counters 2 and 3	RW	1'b0
[1]	pmevcnt01_combined	Enables combination of PMU counters 0 and 1	RW	1'b0
[0]	pmu_en	DTM PMU enable NOTE: All other fields in this register are valid only if this bit is set.	RW	1'b0

4.3.13.44 por_dtm_pmevcnt

Contains all PMU event counters (0, 1, 2, 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2220`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-539: por_dtm_pmevcnt

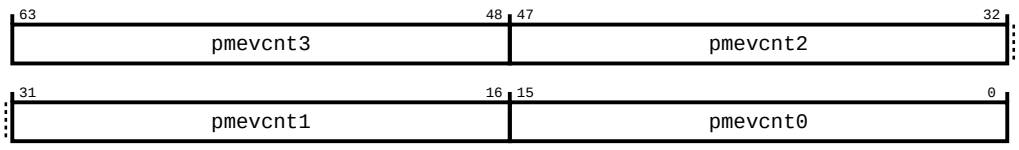


Table 4-555: por_dtm_pmevcnt attributes

Bits	Name	Description	Type	Reset
[63:48]	pmevcnt3	PMU event counter 3	RW	16'h0000
[47:32]	pmevcnt2	PMU event counter 2	RW	16'h0000

Bits	Name	Description	Type	Reset
[31:16]	pmevcnt1	PMU event counter 1	RW	16'h0000
[15:0]	pmevcnt0	PMU event counter 0	RW	16'h0000

4.3.13.45 por_dtm_pmevcntsr

Functions as the PMU event counter shadow register for all counters (0, 1, 2, 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2240`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-540: por_dtm_pmevcntsr

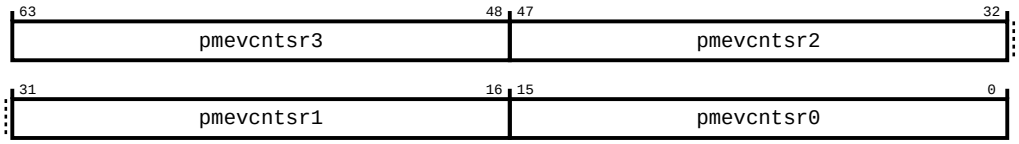


Table 4-556: por_dtm_pmevcntsr attributes

Bits	Name	Description	Type	Reset
[63:48]	pmevcntsr3	PMU event counter 3 shadow register	RW	16'h0000
[47:32]	pmevcntsr2	PMU event counter 2 shadow register	RW	16'h0000
[31:16]	pmevcntsr1	PMU event counter 1 shadow register	RW	16'h0000
[15:0]	pmevcntsr0	PMU event counter 0 shadow register	RW	16'h0000

4.3.13.46 por_dtm_control_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the DTM control register. NOTE: There will be max. of 3 DTM registers based on MXP_MULTIPLE_DTM_EN_PARAM and MXP_NUM_DEV_PORT_PARAM value. Each successive DTM register will be at the next 'h200 + 8 byte address boundary. Each successive DTM register will be named with the suffix corresponding to the DT register number. For example por_dtm_control_dt<0:3>

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2100 + #{512*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-541: por_dtm_control_dt1-3

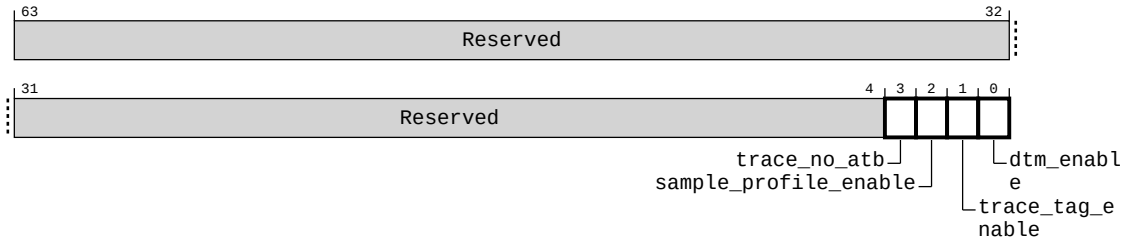


Table 4-557: por_dtm_control_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	trace_no_atb	When set, trace packet is not delivered out of ATB, and FIFO entry holds the first trace packet	RW	1'b0
[2]	sample_profile_enable	Enables sample profile function	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	trace_tag_enable	Watchpoint trace tag enable 1'b1: Trace tag enabled 1'b0: No trace tag	RW	1'b0
[0]	dtm_enable	Enables debug watchpoint and PMU function; prior to writing this bit, all other DT configuration registers must be programmed; once this bit is set, other DT configuration registers must not be modified	RW	1'b0

4.3.13.47 por_dtm_fifo_entry_ready_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Controls status of DTM FIFO entries.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2118 + #{512*index}`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-542: por_dtm_fifo_entry_ready_dt1-3

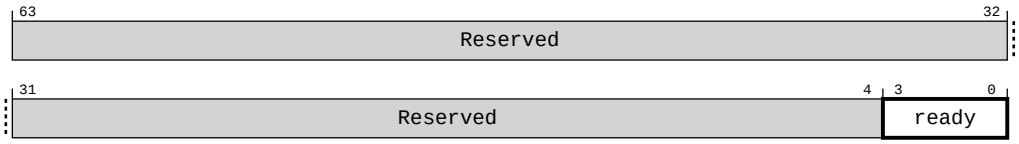


Table 4-558: por_dtm_fifo_entry_ready_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	ready	Indicates which DTM FIFO entries are ready; write a 1 to clear Bit [3]: Entry 3 ready when set Bit [2]: Entry 2 ready when set Bit [1]: Entry 1 ready when set Bit [0]: Entry 0 ready when set	W1C	4'b0

4.3.13.48 por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{index%4} data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2120 + #{24*(index%4)} + #{512*((index/4)+1)}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-543: por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1

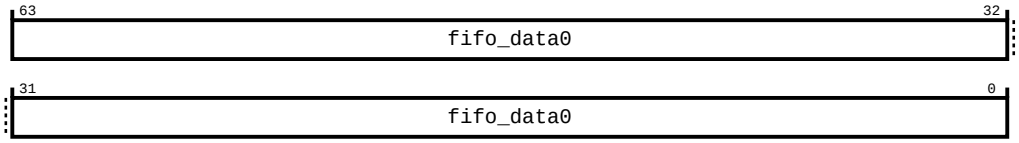


Table 4-559: por_dtm_fifo_entry0-11%4_0_dt(0-11/4)+1 attributes

Bits	Name	Description	Type	Reset
[63:0]	fifo_data0	Entry data bit vector 63:0	RO	64'b0

4.3.13.49 por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{index%4} data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2128 + #{24*(index%4)} + #{512*((index/4)+1)}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-544: por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1

Table 4-560: por_dtm_fifo_entry0-11%4_1_dt(0-11/4)+1 attributes

Bits	Name	Description	Type	Reset
[63:0]	fifo_data1	Entry data bit vector 127:64	RO	64'b0

4.3.13.50 por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Contains DTM FIFO entry #{index%4} data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2130 + #{24*(index%4)} + #{512*((index/4)+1)}`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-545: por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1

Table 4-561: por_dtm_fifo_entry0-11%4_2_dt(0-11/4)+1 attributes

Bits	Name	Description	Type	Reset
[63:48]	fifo_cycle_count	Entry cycle count bit vector 15:0	RO	16'b0
[47:0]	fifo_data2	Entry data bit vector 143:128	RO	48'b0

4.3.13.51 por_dtm_wp0-11%4_config_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Configures watchpoint #{index%4}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h21A0 + #{24*(index%4)} + #{512*((index/4)+1)}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-546: por_dtm_wp0-11%4_config_dt(0-11/4)+1

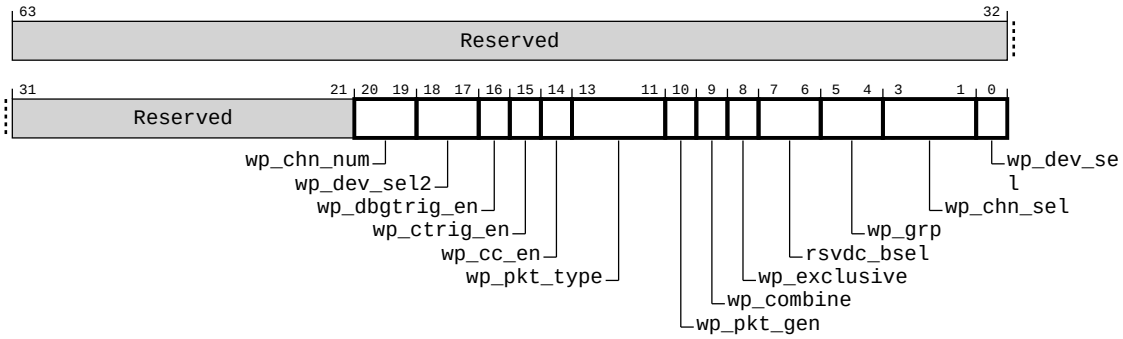


Table 4-562: por_dtm_wp0-11%4_config_dt(0-11/4)+1 attributes

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[20:19]	wp_chn_num	VC number index for replicated channels in specified SMXP	RW	2'b0
[18:17]	wp_dev_sel2	Upper bits for device port selection in specified SMXP	RW	2'b0
[16]	wp_dbgtrig_en	Enables watchpoint debug trigger packet generation	RW	1'b0
[15]	wp_ctrig_en	Enables watchpoint cross trigger packet generation	RW	1'b0
[14]	wp_cc_en	Enables inclusion of cycle count in watchpoint track packet generation	RW	1'b0
[13:11]	wp_pkt_type	Trace packet type 3'b000: TXNID (up to X18) 3'b001: TXNID + opcode (up to X9) 3'b010: TXNID + opcode + source ID + target ID (up to X4) 3'b011: Reserved 3'b100: Control flit 3'b101: DAT flit DATA [127:0] 3'b110: DAT flit DATA [255:128] 3'b111: Reserved	RW	3'b000
[10]	wp_pkt_gen	Enables watchpoint trace packet generation	RW	1'b0
[9]	wp_combine	Enables combination of watchpoints #{index%4} and #{(index%4)+1}	RW	1'b0
[8]	wp_exclusive	Watchpoint mode 1'b0: Regular mode 1'b1: Exclusive mode	RW	1'b0
[7:6]	rsvdc_bsel	Byte select of RSVDC in trace packet 2'h0: Select RSVDC[7:0] 2'h1: Select RSVDC[15:8] 2'h2: Select RSVDC[23:16] 2'h3: Select RSVDC[31:24]	RW	1'b0
[5:4]	wp_grp	Watchpoint register format group 2'h0: Select primary group 2'h1: Select secondary group 2'h2: Select tertiary group 2'h3: Reserved	RW	1'b0
[3:1]	wp_chn_sel	VC selection 3'b000: Select REQ VC 3'b001: Select RSP VC 3'b010: Select SNP VC 3'b011: Select DATA VC NOTE: All other values are reserved.	RW	3'b000
[0]	wp_dev_sel	Device port selection in specified SMXP 1'b0: Select device port 0 1'b1: Select device port 1	RW	1'b0

4.3.13.52 por_dtm_wp0-11%4_val_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Configures watchpoint #{index%4} comparison value.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h21A8 + #{24*(index%4)} + #{512*((index/4)+1)}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-547: por_dtm_wp0-11%4_val_dt(0-11/4)+1

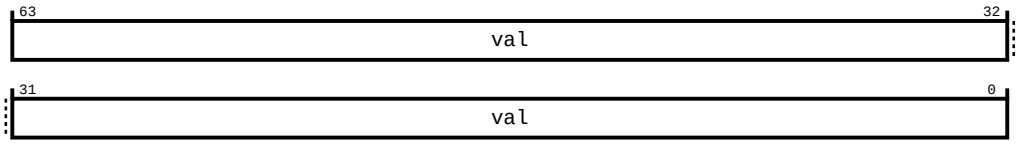


Table 4-563: por_dtm_wp0-11%4_val_dt(0-11/4)+1 attributes

Bits	Name	Description	Type	Reset
[63:0]	val	Refer to DTM watchpoint section for details	RW	64'b0

4.3.13.53 por_dtm_wp0-11%4_mask_dt(0-11/4)+1

There are 12 iterations of this register. The index ranges from 0 to 11. Configures watchpoint #{index%4} comparison mask.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h21B0 + #{24*(index%4)} + #{512*((index/4)+1)}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-548: por_dtm_wp0-11%4_mask_dt(0-11/4)+1

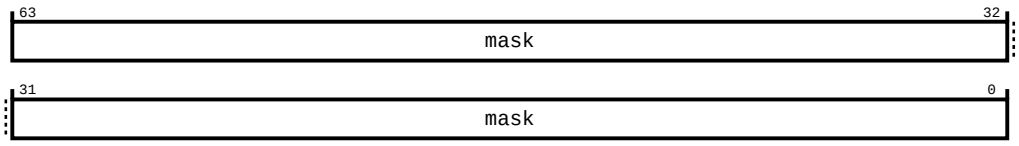


Table 4-564: por_dtm_wp0-11%4_mask_dt(0-11/4)+1 attributes

Bits	Name	Description	Type	Reset
[63:0]	mask	Refer to DTM watchpoint section for details	RW	64'b0

4.3.13.54 por_dtm_pmsicr_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the sampling interval counter register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h2200 + #{512*index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-549: por_dtm_pmsicr_dt1-3

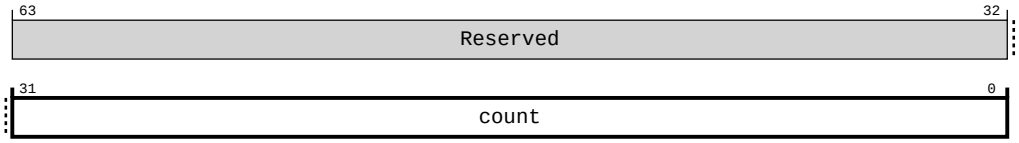


Table 4-565: por_dtm_pmsicr_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	count	Current value of sample counter	RW	32'b0

4.3.13.55 por_dtm_pmsirr_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the sampling interval reload register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2208 + #{512*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-550: por_dtm_pmsirr_dt1-3

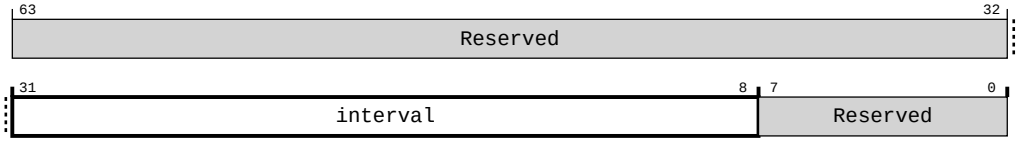


Table 4-566: por_dtm_pmsirr_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:8]	interval	Sampling interval to be reloaded	RW	24'b0
[7:0]	Reserved	Reserved	RO	-

4.3.13.56 por_dtm_pmu_config_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Configures the DTM PMU.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2210 + #{512*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-551: por_dtm_pmu_config_dt1-3

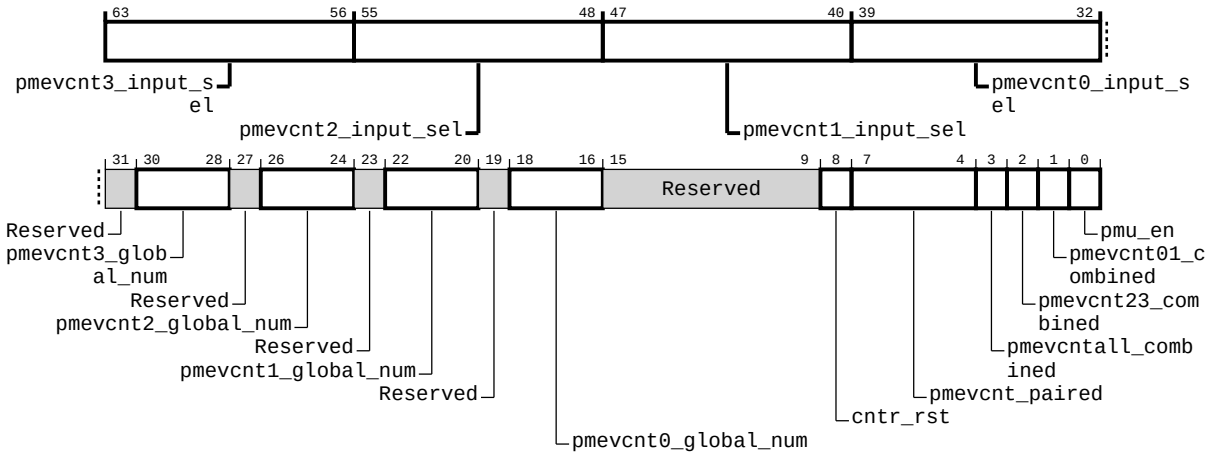


Table 4-567: por_dtm_pmu_config_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:56]	pmevcnt3_input_sel	Source to be counted in PMU counter 3; see pmevcnt0_input_sel for encodings	RW	8'b0
[55:48]	pmevcnt2_input_sel	Source to be counted in PMU counter 2; see pmevcnt0_input_sel for encodings	RW	8'b0
[47:40]	pmevcnt1_input_sel	Source to be counted in PMU counter 1; see pmevcnt0_input_sel for encodings	RW	8'b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	Source to be counted in PMU counter 0: Supports 2 Ports (DT1: P2 and P3, DT2: P4 and P5) when (MXP_NUM_DEV_PORT_PARAM > 2 and MXP_MULTIPLE_DTM_EN_PARAM = 1) **8'h00** Watchpoint 0 **8'h01** Watchpoint 1 **8'h02** Watchpoint 2 **8'h03** Watchpoint 3 **8'h04** XP PMU Event 0 **8'h05** XP PMU Event 1 **8'h06** XP PMU Event 2 **8'h07** XP PMU Event 3 **8'h10** Port 0 Device 0 PMU Event 0 **8'h11** Port 0 Device 0 PMU Event 1 **8'h12** Port 0 Device 0 PMU Event 2 **8'h13** Port 0 Device 0 PMU Event 3 **8'h14** Port 0 Device 1 PMU Event 0 **8'h15** Port 0 Device 1 PMU Event 1 **8'h16** Port 0 Device 1 PMU Event 2 **8'h17** Port 0 Device 1 PMU Event 3 **8'h18** Port 0 Device 2 PMU Event 0 **8'h19** Port 0 Device 2 PMU Event 1 **8'h1A** Port 0 Device 2 PMU Event 2 **8'h1B** Port 0 Device 2 PMU Event 3 **8'h1C** Port 0 Device 3 PMU Event 0 **8'h1D** Port 0 Device 3 PMU Event 1 **8'h1E** Port 0 Device 3 PMU Event 2 **8'h1F** Port 0 Device 3 PMU Event 3 **8'h20** Port 1 Device 0 PMU Event 0 **8'h21** Port 1 Device 0 PMU Event 1 **8'h22** Port 1 Device 0 PMU Event 2 **8'h23** Port 1 Device 0 PMU Event 3 **8'h24** Port 1 Device 1 PMU Event 0 **8'h25** Port 1 Device 1 PMU Event 1 **8'h26** Port 1 Device 1 PMU Event 2 **8'h27** Port 1 Device 1 PMU Event 3 **8'h28** Port 1 Device 2 PMU Event 0 **8'h29** Port 1 Device 2 PMU Event 1 **8'h2A** Port 1 Device 2 PMU Event 2 **8'h2B** Port 1 Device 2 PMU Event 3 **8'h2C** Port 1 Device 3 PMU Event 0 **8'h2D** Port 1 Device 3 PMU Event 1 **8'h2E** Port 1 Device 3 PMU Event 2 **8'h2F** Port 1 Device 3 PMU Event 3	RW	8'b0
[31]	Reserved	Reserved	RO	-
[30:28]	pmevcnt3_global_num	Global counter to pair with PMU counter 3; see pmevcnt0_global_num for encodings	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	pmevcnt2_global_num	Global counter to pair with PMU counter 2; see pmevcnt0_global_num for encodings	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	pmevcnt1_global_num	Global counter to pair with PMU counter 1; see pmevcnt0_global_num for encodings	RW	3'b0
[19]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[18:16]	pmevcnt0_global_num	Global counter to pair with PMU counter 0 **3'b000** Global PMU event counter A **3'b001** Global PMU event counter B **3'b010** Global PMU event counter C **3'b011** Global PMU event counter D **3'b100** Global PMU event counter E **3'b101** Global PMU event counter F **3'b110** Global PMU event counter G **3'b111** Global PMU event counter H	RW	3'b0
[15:9]	Reserved	Reserved	RO	-
[8]	cntr_rst	Enables clearing of live counters upon assertion of snapshot	RW	1'b0
[7:4]	pmevcnt_paired	PMU local counter paired with global counter	RW	4'b0
[3]	pmevcntall_combined	Enables combination of all PMU counters (0, 1, 2, 3) NOTE: When set, pmevcnt01_combined and pmevcnt23_combined have no effect.	RW	1'b0
[2]	pmevcnt23_combined	Enables combination of PMU counters 2 and 3	RW	1'b0
[1]	pmevcnt01_combined	Enables combination of PMU counters 0 and 1	RW	1'b0
[0]	pmu_en	DTM PMU enable NOTE: All other fields in this register are valid only if this bit is set.	RW	1'b0

4.3.13.57 por_dtm_pmevcnt_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Contains all PMU event counters (0, 1, 2, 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'h2220 + #{512*index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-552: por_dtm_pmevcnt_dt1-3

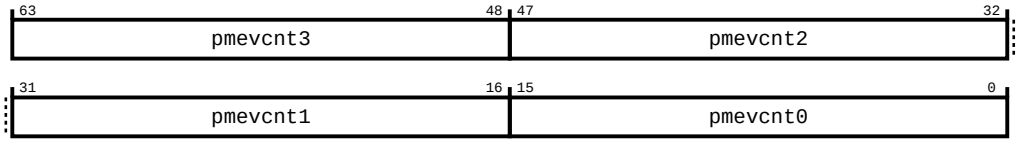


Table 4-568: por_dtm_pmevcnt_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:48]	pmevcnt3	PMU event counter 3	RW	16'h0000
[47:32]	pmevcnt2	PMU event counter 2	RW	16'h0000
[31:16]	pmevcnt1	PMU event counter 1	RW	16'h0000
[15:0]	pmevcnt0	PMU event counter 0	RW	16'h0000

4.3.13.58 por_dtm_pmevcntsr_dt1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Functions as the PMU event counter shadow register for all counters (0, 1, 2, 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2240 + #{512*index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-553: por_dtm_pmevcntsr_dt1-3

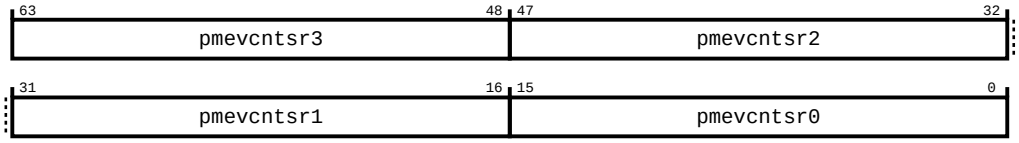


Table 4-569: por_dtm_pmevcntsr_dt1-3 attributes

Bits	Name	Description	Type	Reset
[63:48]	pmevcntsr3	PMU event counter 3 shadow register	RW	16'h0000
[47:32]	pmevcntsr2	PMU event counter 2 shadow register	RW	16'h0000
[31:16]	pmevcntsr1	PMU event counter 1 shadow register	RW	16'h0000
[15:0]	pmevcntsr0	PMU event counter 0 shadow register	RW	16'h0000

4.3.13.59 por_mxp_multi_mesh_chn_sel_0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Functions as the CHI VC channel select per Target register in Multi-Mesh Channel structure.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC00 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.multi_mesh_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-554: por_mxp_multi_mesh_chn_sel_0-15

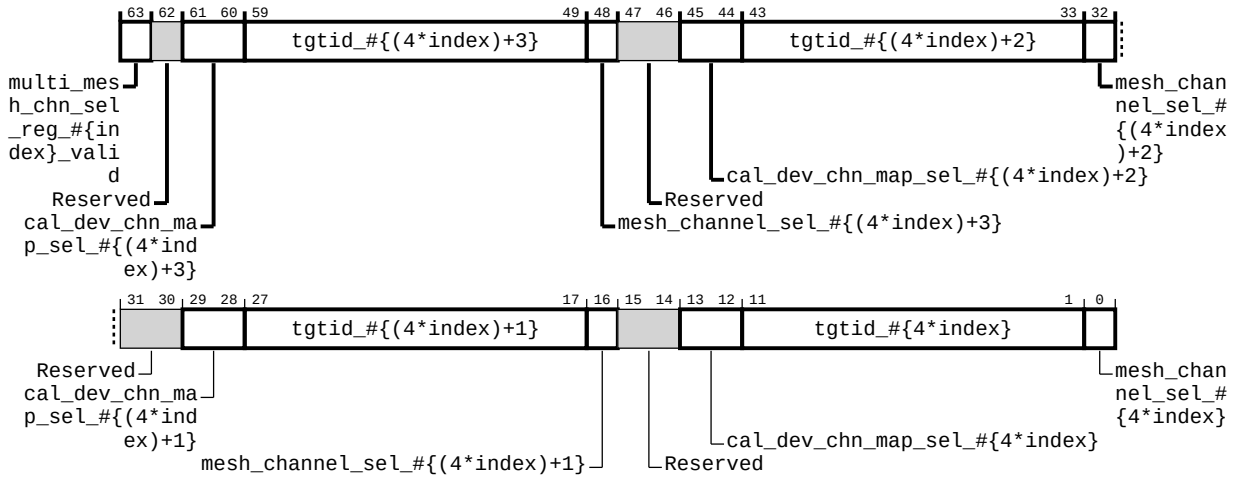


Table 4-570: por_mxp_multi_mesh_chn_sel_0-15 attributes

Bits	Name	Description	Type	Reset
[63]	multi_mesh_chn_sel_reg_#{index}_valid	Indicates that multi mesh CHI VC channel configured for the targets specified in this register.	RW	1'b0
[62]	Reserved	Reserved	RO	-
[61:60]	cal_dev_chn_map_sel_#{(4*index)+3}	Channel Map select for target devices behind CAL (associated with the corresponding tgtid field): **2'b00** CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with > 2 device ports per XP), - CAL2: DEV0,DEV1 are mapped to same channel, **2'b01** CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below, - CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel, **2'b10** Reserved **2'b11** Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below, - CAL2: DEV0,DEV1 can be mapped to different channel, - CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel	RW	2'b0
[59:49]	tgtid_#{(4*index)+3}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	11'b0
[48]	mesh_channel_sel_#{(4*index)+3}	CHI VC channel select: 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	1'b0
[47:46]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[45:44]	cal_dev_chn_map_sel_#{(4*index)+2}	Channel Map select for target devices behind CAL (associated with the corresponding tgtid field): **2'b00** CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with > 2 device ports per XP), - CAL2: DEV0,DEV1 are mapped to same channel, **2'b01** CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below, - CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel, **2'b10** Reserved **2'b11** Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below, - CAL2: DEV0,DEV1 can be mapped to different channel, - CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel	RW	1'b0
[43:33]	tgtid_#{(4*index)+2}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	11'b0
[32]	mesh_channel_sel_#{(4*index)+2}	CHI VC channel select: 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	1'b0
[31:30]	Reserved	Reserved	RO	-
[29:28]	cal_dev_chn_map_sel_#{(4*index)+1}	Channel Map select for target devices behind CAL (associated with the corresponding tgtid field): **2'b00** CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with > 2 device ports per XP), - CAL2: DEV0,DEV1 are mapped to same channel, **2'b01** CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below, - CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel, **2'b10** Reserved **2'b11** Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below, - CAL2: DEV0,DEV1 can be mapped to different channel, - CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel	RW	2'b0
[27:17]	tgtid_#{(4*index)+1}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	11'b0
[16]	mesh_channel_sel_#{(4*index)+1}	CHI VC channel select: 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	1'b0
[15:14]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[13:12]	cal_dev_chn_map_sel_#{4*index}	Channel Map select for target devices behind CAL (associated with the corresponding tgtid field): **2'b00** CAL2: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below (applicable for Mesh with > 2 device ports per XP), - CAL2: DEV0,DEV1 are mapped to same channel, **2'b01** CAL4: All devices behind CAL are mapped to same channel in the Multi-Channel Mesh structure as specified below, - CAL4: DEV0,DEV1,DEV2,DEV3 are mapped to same channel, **2'b10** Reserved **2'b11** Each target device behind CAL can be mapped to different channel in the Multi-Channel Mesh structure as specified below, - CAL2: DEV0,DEV1 can be mapped to different channel, - CAL4: DEV0,DEV1,DEV2,DEV3 can be mapped to different channel	RW	2'b0
[11:1]	tgtid_#{4*index}	11-bit Target ID associated with the corresponding channel_sel field. This field is used in the LUP to determine which CHI VC channel the FLIT has to be routed to for this target.	RW	11'b0
[0]	mesh_channel_sel_#{4*index}	CHI VC channel select: 1 - CHI VC channel 1 is selected, 0 - CHI VC channel 0 is selected	RW	1'b0

4.3.13.60 por_mxp_multi_mesh_chn_ctrl

Functions as the control register for Target based channel selection in Multi-Mesh Channel structure.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC80`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.multi_mesh_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-555: por_mxp_multi_mesh_chn_ctrl

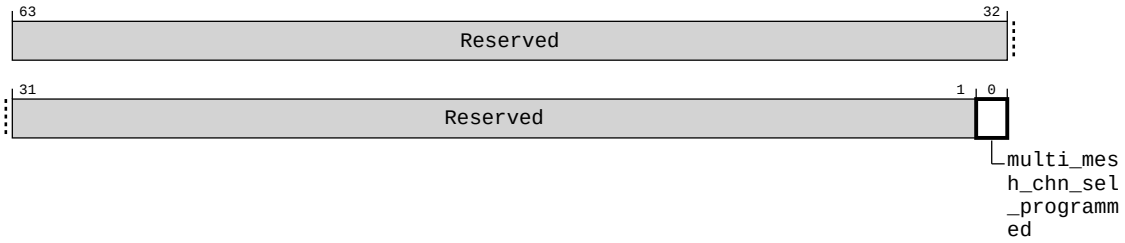


Table 4-571: por_mxp_multi_mesh_chn_ctrl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	multi_mesh_chn_sel_programmed	Indicates that multi CHI VC channel configured for all the targets specified in the channel select registers.	RW	1'b0

4.3.13.61 por_mxp_xy_override_sel_0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Functions as SRC-TGT pair whose X-Y route path can be overriden (for Non-XY Route feature per Souce-Target pair) per XP.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hC90 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.xy_override_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-556: por_mxp_xy_override_sel_0-7

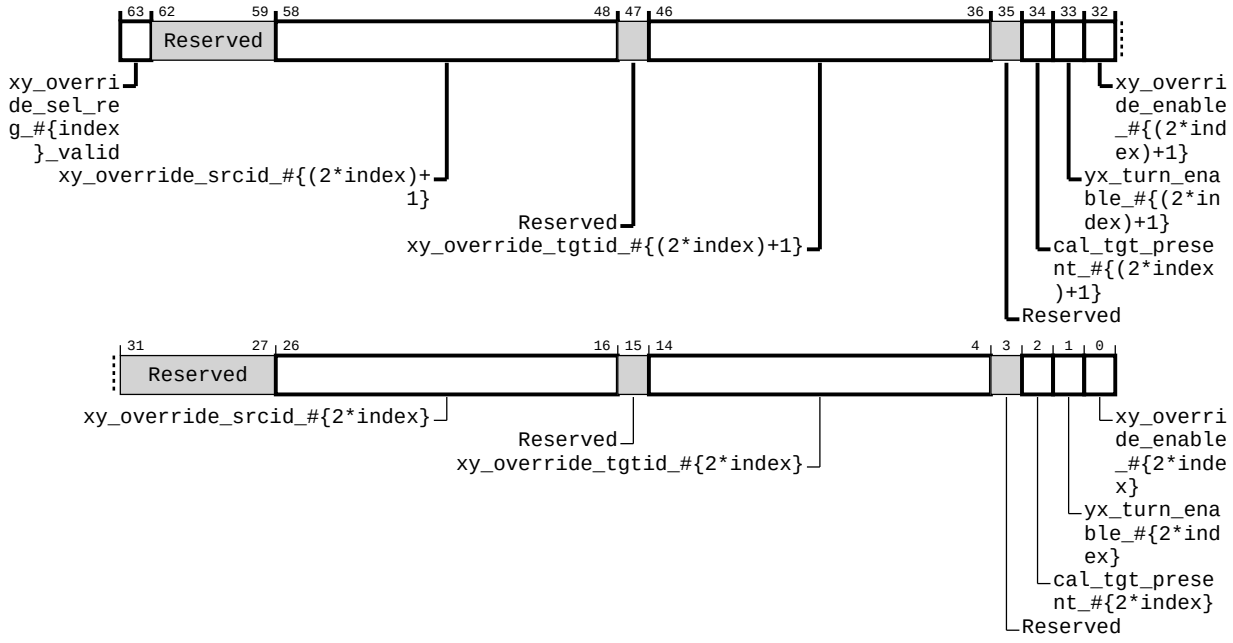


Table 4-572: por_mxp_xy_override_sel_0-7 attributes

Bits	Name	Description	Type	Reset
[63]	xy_override_sel_reg_#{index}_valid	Indicates that Source-Target pairs whose X-Y route path can be overriden are configured in this register.	RW	1'b0
[62:59]	Reserved	Reserved	RO	-
[58:48]	xy_override_srcid_#{(2*index)+1}	11-bit Source ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overriden.	RW	11'b0
[47]	Reserved	Reserved	RO	-
[46:36]	xy_override_tgtid_#{(2*index)+1}	11-bit Target ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overriden.	RW	11'b0
[35]	Reserved	Reserved	RO	-
[34]	cal_tgt_present_#{(2*index)+1}	CAL TGT Presence Indication for XY Route Override of all devices behind CAL: 1 - CAL4 TGT Present for XY Route Override of all devices behind CAL, 0 - CAL2 TGT or no CAL TGT Present for XY Route Override of all devices behind CAL.	RW	1'b0
[33]	yx_turn_enable_#{(2*index)+1}	Y-X Turn Enable: 1 - Y-X Turn enabled for associated Source-Target Pair, 0 - Y-X Turn disabled	RW	1'b0
[32]	xy_override_enable_#{(2*index)+1}	X-Y Route Override Enable: 1 - X-Y Route override enabled for associated Source-Target Pair, 0 - X-Y Route override disabled	RW	1'b0
[31:27]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[26:16]	xy_override_srcid_#{2*index}	11-bit Source ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overriden.	RW	11'b0
[15]	Reserved	Reserved	RO	-
[14:4]	xy_override_tgtid_#{2*index}	11-bit Target ID associated with the XY Override. This field is used in the LUP to determine if XY route for the associated source-target pair needs to be overriden.	RW	11'b0
[3]	Reserved	Reserved	RO	-
[2]	cal_tgt_present_#{2*index}	CAL TGT Presence Indication for XY Route Override of all devices behind CAL: 1 - CAL4 TGT Present for XY Route Override of all devices behind CAL, 0 - CAL2 TGT or no CAL TGT Present for XY Route Override of all devices behind CAL.	RW	1'b0
[1]	yx_turn_enable_#{2*index}	Y-X Turn Enable: 1 - Y-X Turn enabled for associated Source-Target Pair, 0 - Y-X Turn disabled	RW	1'b0
[0]	xy_override_enable_#{2*index}	X-Y Route Override Enable: 1 - X-Y Route override enabled for associated Source-Target Pair, 0 - X-Y Route override disabled	RW	1'b0

4.3.13.62 por_mxp_p0-5_pa2setaddr_slc

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the control register of PA to SetAddr and vice versa conversion for HNF-SLC on XP port #{index}. NOTE: There will be max. of 6 MXP Port registers based on MXP_NUM_DEV_PORT_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCD0 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-557: por_mxp_p0-5_pa2setaddr_slc

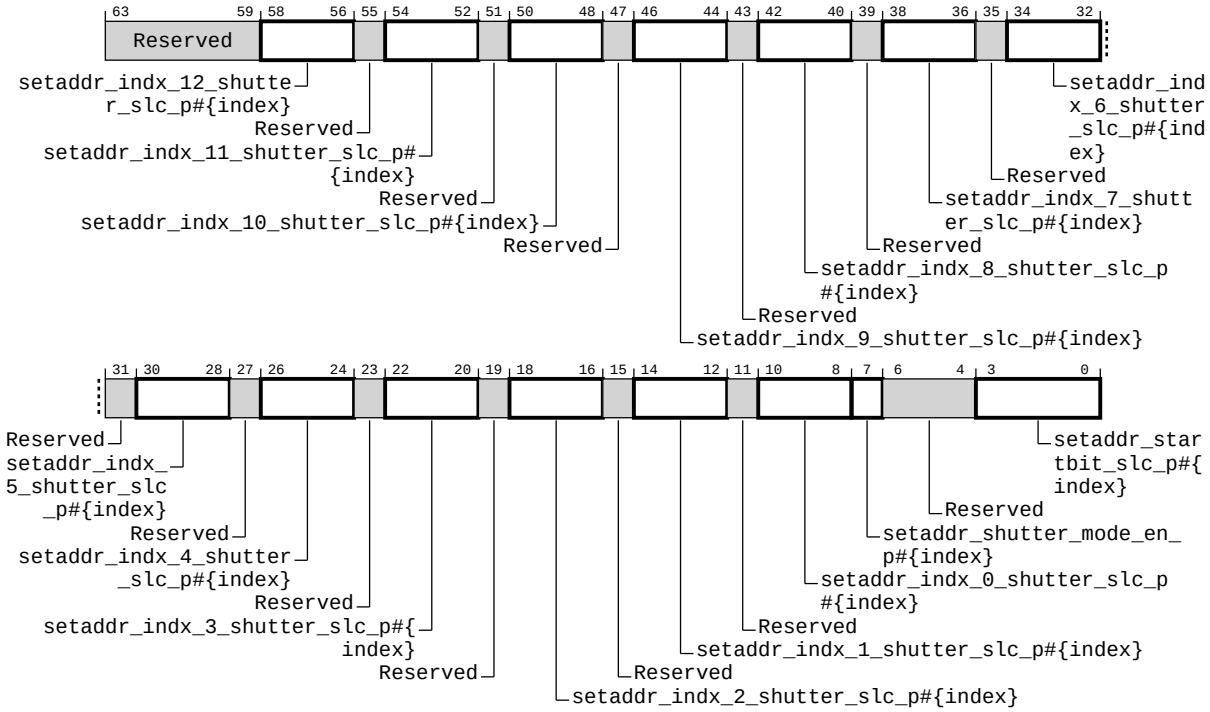


Table 4-573: por_mxp_p0-5_pa2setaddr_slc attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:56]	setaddr_indx_12_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_indx_11_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0

Bits	Name	Description	Type	Reset
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_indx_10_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_indx_9_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_indx_8_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_indx_7_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[35]	Reserved	Reserved	RO	-
[34:32]	setaddr_indx_6_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[31]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[30:28]	setaddr_indx_5_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_indx_4_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_indx_3_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[19]	Reserved	Reserved	RO	-
[18:16]	setaddr_indx_2_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_indx_1_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[11]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[10:8]	setaddr_indx_0_shutter_slc_p#{index}	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_slc **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[7]	setaddr_shutter_mode_en_p#{index}	Enables address shuttering mode for SLC as programmed by setaddr_indx_X_shutter registers	RW	1'b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_slc_p#{index}	SLC: SetAddr starting bit for SLC in HNF connected to port p#{index} **4'b0110** Setaddr starts from PA[6] **4'b0111** Setaddr starts from PA[7] **4'b1000** Setaddr starts from PA[8] **4'b1001** Setaddr starts from PA[9] **4'b1010** Setaddr starts from PA[10] **4'b1011** Setaddr starts from PA[11] **4'b1100** Setaddr starts from PA[12]	RW	4'b0110

4.3.13.63 por_mxp_p0-5_pa2setaddr_sf

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the control register of PA to Set/TagAddr and vice versa conversion for HNF-SF on XP port #{index}. NOTE: There will be max. of 6 MXP Port registers based on MXP_NUM_DEV_PORT_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCD8 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-558: por_mxp_p0-5_pa2setaddr_sf

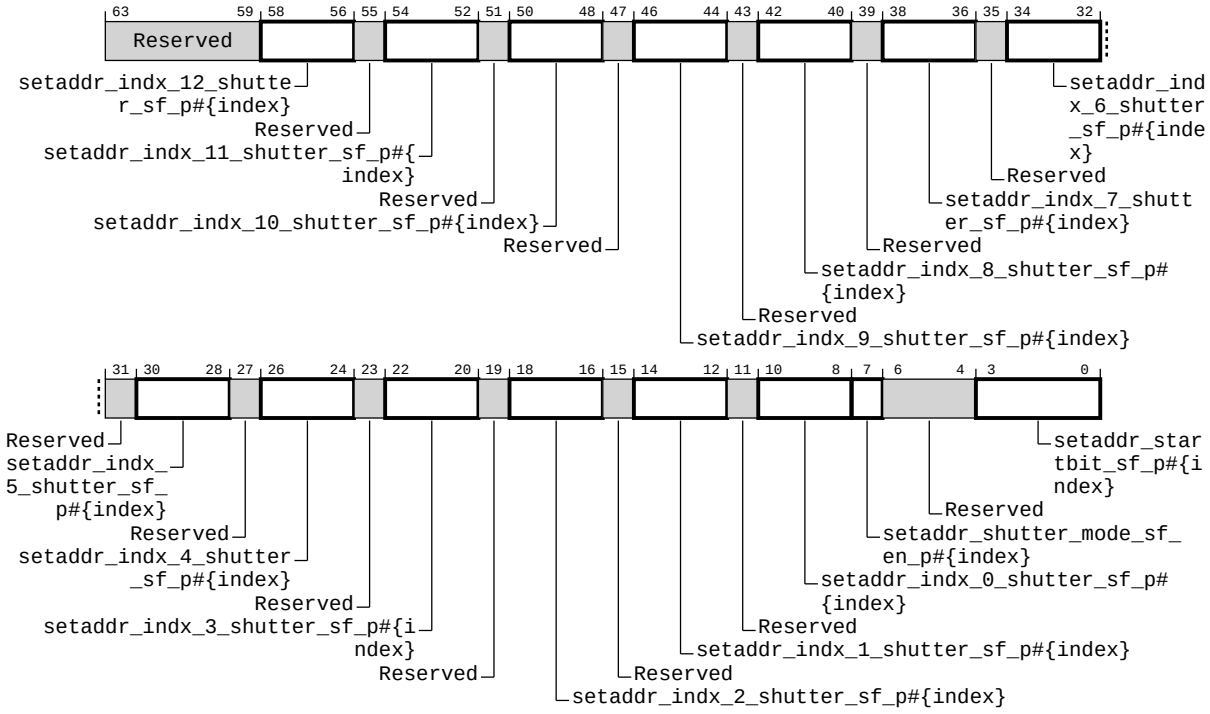


Table 4-574: por_mxp_p0-5_pa2setaddr_sf attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:56]	setaddr_indx_12_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[55]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[54:52]	setaddr_indx_11_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_indx_10_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_indx_9_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_indx_8_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_indx_7_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[35]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[34:32]	setaddr_indx_6_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_indx_5_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_indx_4_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_indx_3_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[19]	Reserved	Reserved	RO	-
[18:16]	setaddr_indx_2_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:12]	setaddr_indx_1_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[11]	Reserved	Reserved	RO	-
[10:8]	setaddr_indx_0_shutter_sf_p#{index}	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_sf **3'b000** pass-through **3'b001** shift_1 **3'b010** shift_2 **3'b011** shift_3 **3'b100** shift_4 **3'b101** shift_5	RW	3'b0
[7]	setaddr_shutter_mode_sf_en_p#{index}	Enables address shuttering mode for SF as programmed by setaddr_indx_X_shutter_sf registers	RW	1'b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_sf_p#{index}	SF: SetAddr starting bit for SF in HNF connected to port p#{index} **4'b0110** Setaddr starts from PA[6] **4'b0111** Setaddr starts from PA[7] **4'b1000** Setaddr starts from PA[8] **4'b1001** Setaddr starts from PA[9] **4'b1010** Setaddr starts from PA[10] **4'b1011** Setaddr starts from PA[11] **4'b1100** Setaddr starts from PA[12]	RW	4'b0110

4.3.13.64 por_mxp_p0-5_pa2setaddr_flex_slc

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the SLC control register of PA to Set/TagAddr and vice versa conversion for HNF (flexible) on XP port #{index}. NOTE: There will be max. of 6 MXP Port registers based on MXP_NUM_DEV_PORT_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'hCE0 + #{32*index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-559: por_mxp_p0-5_pa2setaddr_flex_slc

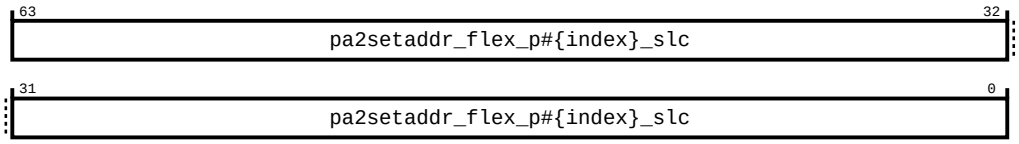


Table 4-575: por_mxp_p0-5_pa2setaddr_flex_slc attributes

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_p#{index}_slc	FLEXIBLE: PA to SET/TAG ADDR and vice versa conversion config field for HNF connected to port p#{index}	RW	64'b0

4.3.13.65 por_mxp_p0-5_pa2setaddr_flex_sf

There are 6 iterations of this register. The index ranges from 0 to 5. Functions as the SF control register of PA to Set/TagAddr and vice versa conversion for HNF (flexible) on XP port #{index}. NOTE: There will be max. of 6 MXP Port registers based on MXP_NUM_DEV_PORT_PARAM value. Each successive MXP Port register will be at the next 8 byte address boundary.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hCE8 + #{32*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.pa2setaddr_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-560: por_mxp_p0-5_pa2setaddr_flex_sf

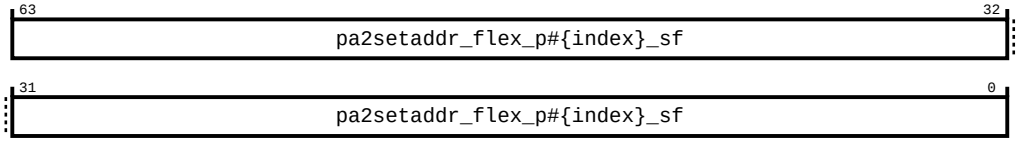


Table 4-576: por_mxp_p0-5_pa2setaddr_flex_sf attributes

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_p#{index}_sf	FLEXIBLE: PA to SET/TAG ADDR conversion and vice versa config field for HNF connected to port p#{index}	RW	64'b0

4.3.14 RN-D register descriptions

This section lists the RN-D registers.

4.3.14.1 por_rnd_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-561: por_rnd_node_info

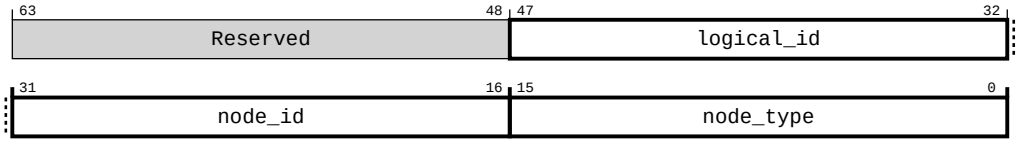


Table 4-577: por_rnd_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h000D

4.3.14.2 por_rnd_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-562: por_rnd_child_info

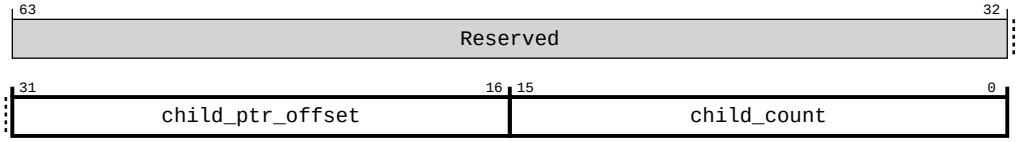


Table 4-578: por_rnd_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'h0

4.3.14.3 por_rnd_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-563: por_rnd_secure_register_groups_override

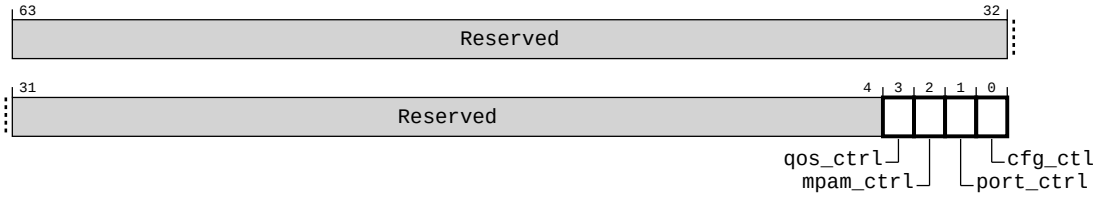


Table 4-579: por_rnd_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	qos_ctrl	Allows Non-secure access to Secure QoS control registers	RW	1'b0
[2]	mpam_ctrl	Allows Non-secure access to Secure AXI port MPAM override register	RW	1'b0
[1]	port_ctrl	Allows Non-secure access to Secure AXI port control registers	RW	1'b0
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register	RW	1'b0

4.3.14.4 por_rnd_unit_info

Provides component identification information for RN-D.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-564: por_rnd_unit_info

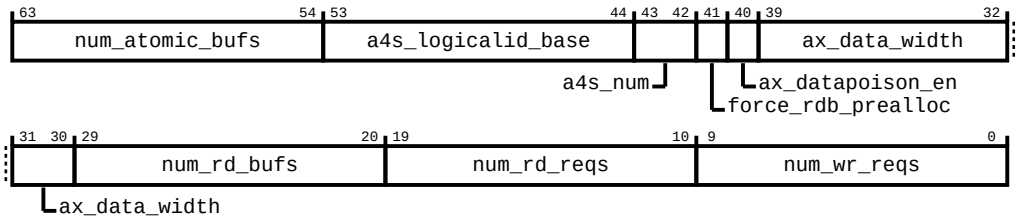


Table 4-580: por_rnd_unit_info attributes

Bits	Name	Description	Type	Reset
[63:54]	num_atomic_bufs	Number of atomic data buffers	RO	Configuration dependent
[53:44]	a4s_logicalid_base	AXI4Stream interfaces logical ID base	RO	Configuration dependent
[43:42]	a4s_num	Number of AXI4Stream interfaces present	RO	Configuration dependent
[41]	force_rdb_prealloc	Force read data buffer preallocation **1'b1** Yes **1'b0** No	RO	Configuration dependent
[40]	ax_datapoison_en	Data Poison enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[39:30]	ax_data_width	AXI interface data width in bits	RO	Configuration dependent
[29:20]	num_rd_bufs	Number of read data buffers	RO	Configuration dependent
[19:10]	num_rd_reqs	Number of outstanding read requests	RO	Configuration dependent
[9:0]	num_wr_reqs	Number of outstanding write requests	RO	Configuration dependent

4.3.14.5 por_rnd_unit_info2

Provides additonal component identification information for RN-D.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-565: por_rnd_unit_info2

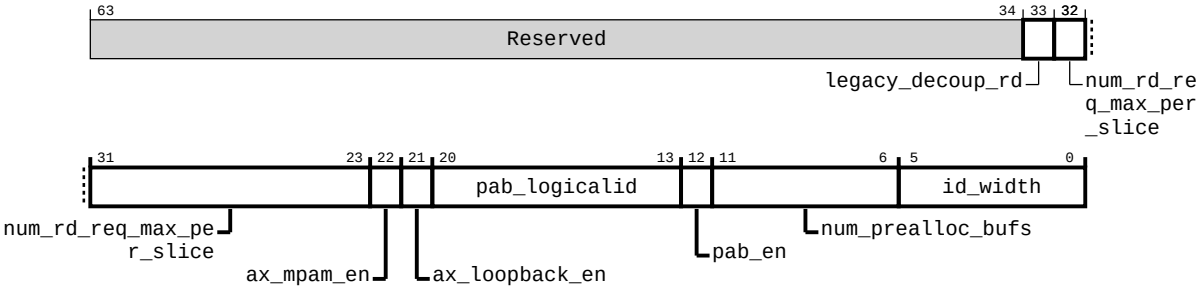


Table 4-581: por_rnd_unit_info2 attributes

Bits	Name	Description	Type	Reset
[63:34]	Reserved	Reserved	RO	-
[33]	legacy_decoup_rd	Legacy decoupled read mode, no read burst propagation	RO	Configuration dependent
[32:23]	num_rd_req_max_per_slice	Number of read request entires per slice	RO	Configuration dependent
[22]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[21]	ax_loopback_en	LoopBack enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[20:13]	pab_logicalid	PUB AUB bridge Logical ID	RO	Configuration dependent
[12]	pab_en	PUB AUB bridge enable **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[11:6]	num_prealloc_bufs	Number of Pre-allocated Read Data Buffers	RO	Configuration dependent
[5:0]	id_width	AXI ID width for ACE-Lite subordinate ports	RO	Configuration dependent

4.3.14.6 por_rnd_cfg_ctl

Functions as the configuration control register. Specifies the current mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA00

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-566: por_rnd_cfg_ctl

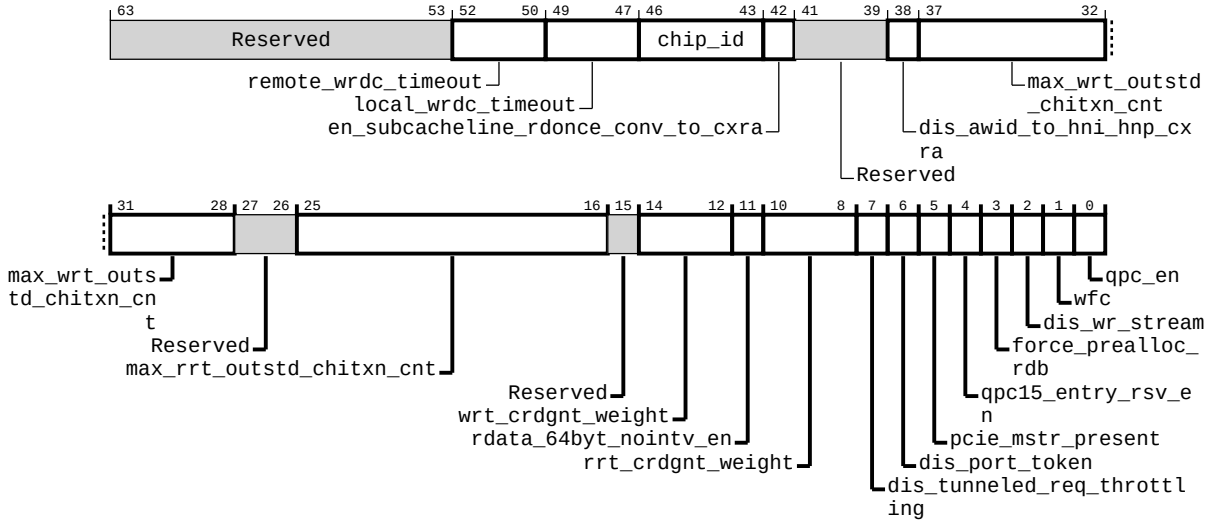


Table 4-582: por_rnd_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:53]	Reserved	Reserved	RO	-
[52:50]	remote_wrdc_timeout	Configurable write data cancel timeout value for remote traffic. **3'b000** ~2000 cycles **3'b001** ~4000 cycles **3'b010** ~8000 cylces **3'b011** ~16000 cycles (default) **3'b100** ~32000 cycles (all other values rsvd)	RW	3'b011
[49:47]	local_wrdc_timeout	Configurable write data cancel timeout value for local traffic. **3'b000** ~2000 cycles **3'b001** ~4000 cycles **3'b010** ~8000 cylces **3'b011** ~16000 cycles (default) **3'b100** ~32000 cycles (all other values rsvd)	RW	3'b010
[46:43]	chip_id	Configurable ChipID for this RNX instance. Must be correctly set for proper handling of remote traffic to HNI/HNP. Only supports values 0..3. Two MSB's are reserved.	RW	4'b0000
[42]	en_subcacheline_rdonce_conv_to_cxra	If set, enables the conversion of sub-cacheline RdOnce to RdNoSnp for CXRA targets	RW	1'b0
[41:39]	Reserved	Reserved	RO	-
[38]	dis_awid_to_hni_hnp_cxra	If set, disables compressed AWID to HNI and CXRA, also disables compressed AWID based ordering. Set this bit if uniq-ID write performance is needed.	RW	1'b0
[37:28]	max_wrt_outstd_chitxn_cnt	Maximum number of outstanding writes allowed on CHI-side	RW	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	max_rrt_outstd_chitxn_cnt	Maximum number of outstanding reads allowed on CHI-side	RW	Configuration dependent
[15]	Reserved	Reserved	RO	-
[14:12]	wrt_crdgnt_weight	Determines weight of credit grant allocated to retried writes in presence of pending retried reads	RW	3'b001
[11]	rdata_64byt_nointv_en	Enables no interleaving property on normal memory read data within 64B granule when set	RW	1'b1
[10:8]	rrt_crdgnt_weight	Determines weight of credit grant allocated to retried reads in presence of pending retried writes	RW	3'b100
[7]	dis_tunneled_req_throttling	Disables retry based throttling of tunneled write requests	RW	1'b0
[6]	dis_port_token	If set, disables per port reservation in the tracker(rd and wr)	RW	1'b1
[5]	pcie_mstr_present	Indicates PCIe manager is present; must be set if PCIe manager is present upstream of RN-I or RN-D	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	qpc15_entry_rsv_en	Enables QPC15 entry reservation **1'b1** Reserves tracker entry for QoS15 requests **1'b0** Does not reserve tracker entry for QoS15 requests NOTE: Only valid and applicable when por_rnd_qpc_en is set	RW	1'b0
[3]	force_prealloc_rdb	When set, all reads from the RN-D are sent with a preallocated read data buffer	RW	Configuration dependent
[2]	dis_wr_stream	Disables streaming of ordered writes when set	RW	1'b0
[1]	wfc	When set, enables waiting for completion (COMP) before dispatching dependent transaction (TXN)	RW	1'b0
[0]	qpc_en	When set, enables QPC-based scheduling using two QoS priority classes (QoS15 and non-QoS15)	RW	1'b1

4.3.14.7 por_rnd_aux_ctl

Functions as the auxiliary control register for RN-D.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-567: por_rnd_aux_ctl

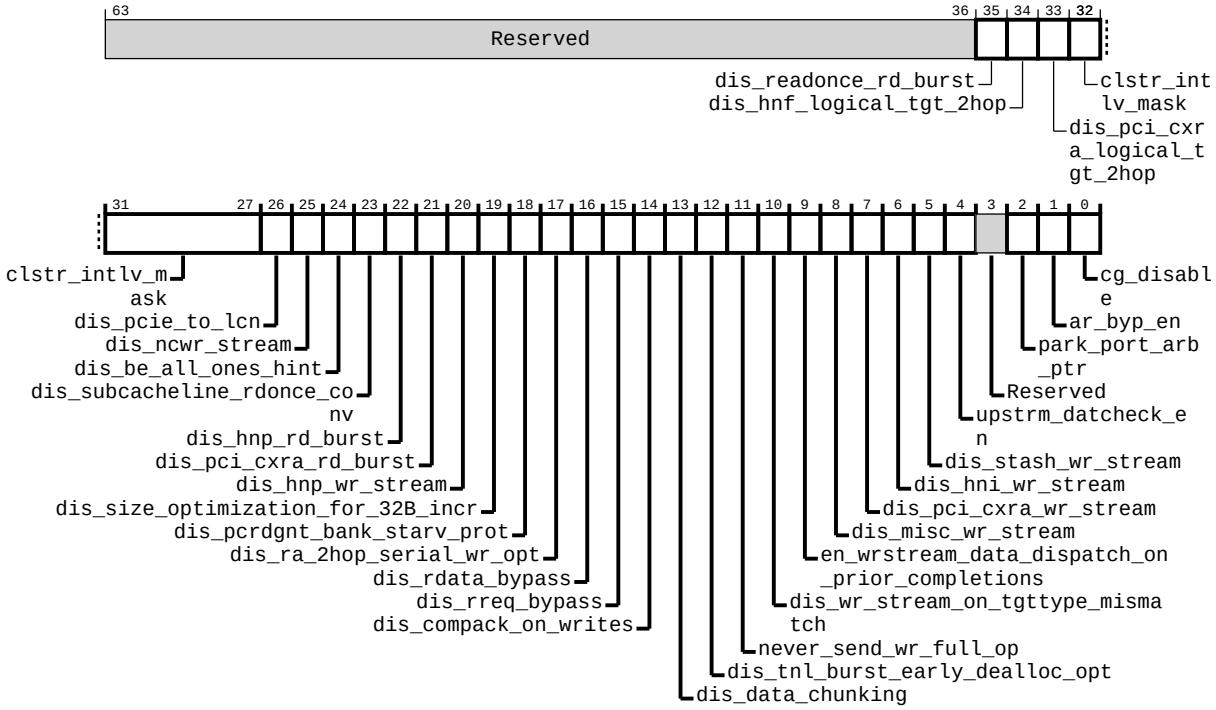


Table 4-583: por_rnd_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[39]	sys_dis_data_interleaving,	System optimized disable read DATA interleaving for all ports. Disables all read data interleaving, including atomic read data being returned for all AXI ports. Read burst preservation is enabled similar to normal mode, but this requires certain system level restrictions: 1. Cannot set SYS_DIS_DATA_INTERLEAVING for multi-chip systems. Support for remote HN-P is a future feature. 2. When setting SYS_DIS_DATA_INTERLEAVING for an RN-I/RN-D it is required for that RN-I/RN-D to target only one HN-P with read bursts. It is also required for the path to a single remote HN-P, when remote support is available, to only go through a single CCG and for the ccg_rni to also have SYS_DIS_DATA_INTERLEAVING set. 3. The AXI subordinate downstream of HN-P must not interleave read burst data. 4. Must have NUM_RD_REQ_PARAM==NUM_RD_BUF_PARAM and NUM_RD_REQ_PRAM<=256, otherwise this bit has no effect. 5. The sum of the maximum number of beats of a cracked burst per each port must fit wholly within an arslice - breaking this rule will result in a hang. Must comprehend DIS_PORT_TOKEN AND QPC15_ENTRY_RSV settings, which will limit number of available entries	RW	1'b0
[38:36]	Reserved	Reserved	RO	-
[35]	dis_readonce_rd_burst	If set, disables read burst for ReadOnce from AXI.	RW	1'b0
[34]	dis_hnf_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical target is pci_cxra and logical target is hnf, otherwise may tunneling/2hop to RA if interleaving granularity settings allow.	RW	1'b0
[33]	dis_pci_cxra_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical and logical target is pci_cxra, otherwise tunneling/2hop to RA.	RW	1'b0
[32:27]	clstr_intlv_mask	Encoded static mask for max interleave granularity supported. When this setting is less than or equal to rnsam's programmed interleave granularity for a write to pci_cxra, tunneling/2hop flow will be used. **6'b111111** 64B **6'b111110** 128B **6'b111100** 256B **6'b111000** 512B **6'b110000** 1024B **6'b100000** 2048B **6'b000000** 4096B **Others** Reserved	RW	6'b000000
[26]	dis_pcie_to_lcn	If set, all pcie traffic sent directly to HNF/CCG, bypasses LCN. Only has effect when pcie_mstr_present	RW	1'b1
[25]	dis_ncwr_stream	Disables streaming of ordered non-cacheable writes when set	RW	1'b0

Bits	Name	Description	Type	Reset
[24]	dis_be_all_ones_hint	If set, disables hint to HNF which signals all BE=1's on writes	RW	1'b0
[23]	dis_subcacheline_rdonce_conv	If set, disables the conversion of sub-cacheline RdOnce to RdNoSnp across all targets	RW	1'b0
[22]	dis_hnp_rd_burst	If set, disables read burst to HNP on CHI request flits . Read burst on CHI is supported only in non-decoupled RDB configuration.	RW	1'b0
[21]	dis_pci_cxra_rd_burst	If set, disables read burst to PCI-CXRA on CHI request flits . Read burst on CHI is supported only in non-decoupled RDB configuration.	RW	1'b0
[20]	dis_hnp_wr_stream	Disables streaming of ordered writes to HNP when set	RW	1'b0
[19]	dis_size_optimization_for_32B_incr	If set, disables the size related optimization for a 32B INCR burst (rh-2512). Only applies to writes.	RW	1'b0
[18]	dis_pcrdgnt_bank_starv_prot	If set, disables across arslice starvation protection	RW	1'b0
[17]	dis_ra_2hop_serial_wr_opt	If set, disables 2 hop indication to ra for serilized writes; will indicate 3 hop	RW	1'b0
[16]	dis_rdata_bypass	If set, disables read data bypass path	RW	1'b0
[15]	dis_rreq_bypass	If set, disables read request bypass path	RW	1'b0
[14]	dis_compack_on_writes	If set, disables comp_ack on streaming writes. WrData is used for ordering writes	RW	1'b1
[13]	dis_data_chunking	If set, disables the data chunking feature	RW	1'b0
[12]	dis_tnl_burst_early_dealloc_opt	If set, disables the optimization related to early deallocation of tunnelled writes for intermediate txns of burst	RW	1'b0
[11]	never_send_wr_full_op	If set, RNI will never send WR FULL op. All write ops will be of PTL type	RW	1'b0
[10]	dis_wr_stream_on_tgttype_mismatch	If set, serializes first write when moving from one tgttype to another	RW	1'b0
[9]	en_wrstream_data_dispatch_on_prior_completions	If set, data dispatch for streaming writes waits for completion of all older writes	RW	1'b0
[8]	dis_misc_wr_stream	Disables streaming of ordered writes with following attributes when set : Device memory or EWA=0 or Excl=1	RW	1'b0
[7]	dis_pci_cxra_wr_stream	Disables streaming of ordered writes to PCI-CXRA when set	RW	1'b0
[6]	dis_hni_wr_stream	Disables streaming of ordered writes to HNI when set	RW	1'b0
[5]	dis_stash_wr_stream	Disables streaming of ordered WrUniqStash when set	RW	1'b0
[4]	upstrm_datcheck_en	Upstream supports Datacheck	RW	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2]	park_port_arb_ptr	Parks the AXI port arbitration pointer for Burst	RW	1'b0
[1]	ar_byp_en	AR bypass enable; enables bypass path in the AR pipeline	RW	1'b1
[0]	cg_disable	Disables clock gating when set	RW	1'b0

4.3.14.8 por_rnd_s0-2_port_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface settings.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA10 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.port_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-568: por_rnd_s0-2_port_control

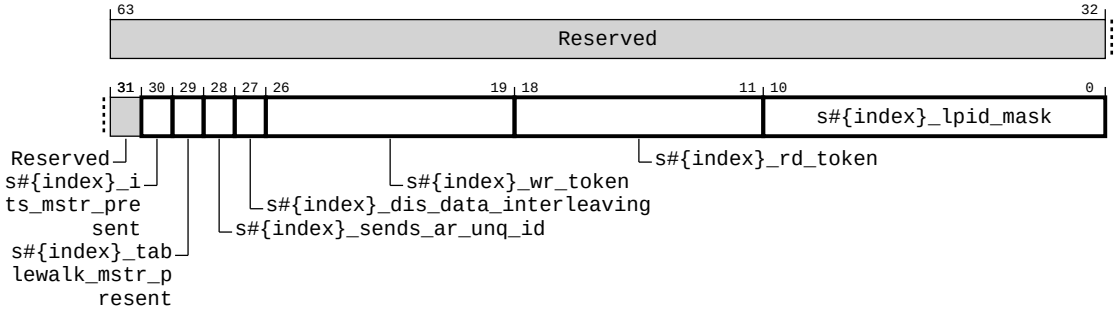


Table 4-584: por_rnd_s0-2_port_control attributes

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[30]	s#{index}_its_mstr_present	Must be set if translation table walk manager present such as TCU or GIC for non-PCIE case. This affects RND AW channel only.	RW	1'b0
[29]	s#{index}_tablewalk_mstr_present	Must be set if translation table walk manager present such as TCU or GIC. This affects RND AR channel only.	RW	1'b0
[28]	s#{index}_sends_ar_unq_id	If set, indicates AR transactions on Port#{index} are always Unique ID. This bit for a port must be set to 1 to enable Read Burst on the CHI side of RND.	RW	1'b0
[27]	s#{index}_dis_data_interleaving	If set, disables read DATA interleaving on RDATAS#{index} channel. This applies only to RDATA generated as a response to requests on AR channel . This does not apply to RDATA generated as a response to Atomic request on AW channel. I.e. RDATA of an Atomic op, on AW channel, may interleave with RDATA of an AR channel request	RW	1'b0
[26:19]	s#{index}_wr_token	Port S#{index} reserved token count for AW channel This must be less than the number of Wr requests(RNID_NUM_WR_REQ_PARAM) on AW achnnel	RW	8'b0000_0000
[18:11]	s#{index}_rd_token	Port S#{index} reserved token count for AR channel per slice This should be less than the number of Rd requests(RNID_NUM_RD_REQ_PARAM) per slice on AR achnnel	RW	8'b0000_0000
[10:0]	s#{index}_lpid_mask	Port S#{index} LPID mask LPID[0]: Equal to the result of UnaryOR of BitwiseAND of LPID mask and AXID (LPID[0] = |(AXID & mask)); specifies which AXID bit is reflected in the LSB of LPID LPID[2:1]: Equal to port ID[1:0]; the MSB of LPID contains port ID	RW	11'b000_0000_0000

4.3.14.9 por_rnd_s0-2_mpam_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface MPAM override values

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA28 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.mpam_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-569: por_rnd_s0-2_mpam_control

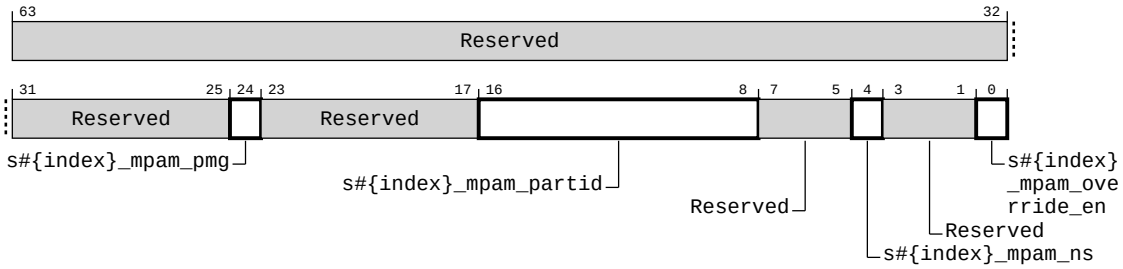


Table 4-585: por_rnd_s0-2_mpam_control attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	s#{index}_mpam_pmg	Port S#{index} MPAM_PMG value	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	s#{index}_mpam_partid	Port S#{index} MPAM_PARTID value	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	s#{index}_mpam_ns	Port S#{index} MPAM_NS value	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	s#{index}_mpam_override_en	Port S#{index} MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Note that when RNID_AXMPAM_EN_PARAM is set to 0, MPAM override value is always used irrespective of this bit value	RW	1'b0

4.3.14.10 por_rnd_s0-2_qos_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS settings for port S#{index} AXI/ACE subordinate interface.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'hA80 + #{index*32}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-570: por_rnd_s0-2_qos_control

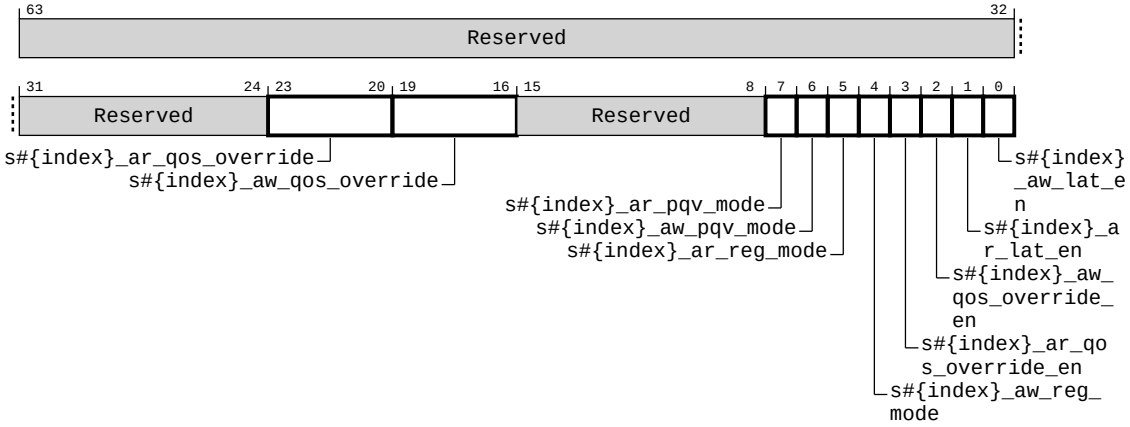


Table 4-586: por_rnd_s0-2_qos_control attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	s#{index}_ar_qos_override	AR QoS override value for port S#{index}	RW	4'b0000
[19:16]	s#{index}_aw_qos_override	AW QoS override value for port S#{index}	RW	4'b0000
[15:8]	Reserved	Reserved	RO	-
[7]	s#{index}_ar_pqv_mode	Configures the QoS regulator mode for read transactions during period mode **1'b0** Normal mode; QoS value is stable when the manager is idle **1'b1** Quiesce high mode; QoS value tends to the maximum value when the manager is idle	RW	1'b0
[6]	s#{index}_aw_pqv_mode	Configures the QoS regulator mode for write transactions during period mode **1'b0** Normal mode; QoS value is stable when the manager is idle **1'b1** Quiesce high mode; QoS value tends to the maximum value when the manager is idle	RW	1'b0

Bits	Name	Description	Type	Reset
[5]	s#{index}_ar_reg_mode	Configures the QoS regulator mode for read transactions **1'b0** Latency mode **1'b1** Period mode; used for bandwidth regulation	RW	1'b0
[4]	s#{index}_aw_reg_mode	Configures the QoS regulator mode for write transactions **1'b0** Latency mode **1'b1** Period mode; used for bandwidth regulation	RW	1'b0
[3]	s#{index}_ar_qos_override_en	Enables port S#{index} AR QoS override; when set, allows QoS value on inbound AR transactions to be overridden	RW	1'b0
[2]	s#{index}_aw_qos_override_en	Enables port S#{index} AW QoS override; when set, allows QoS value on inbound AW transactions to be overridden	RW	1'b0
[1]	s#{index}_ar_lat_en	Enables port S#{index} AR QoS regulation when set	RW	1'b0
[0]	s#{index}_aw_lat_en	Enables port S#{index} AW QoS regulation when set	RW	1'b0

4.3.14.11 por_rnd_s0-2_qos_lat_tgt

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS target latency (in cycles) for regulations of port S#{index} read and write transactions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA88 + #{index*32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-571: por_rnd_s0-2_qos_lat_tgt

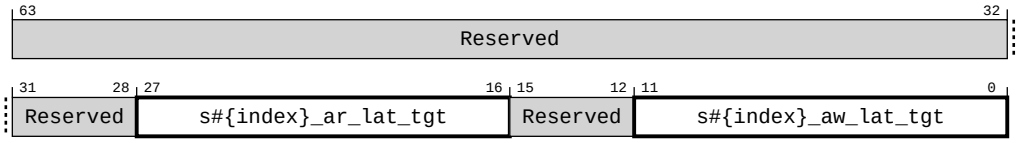


Table 4-587: por_rnd_s0-2_qos_lat_tgt attributes

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:16]	s#{index}_ar_lat_tgt	Port S#{index} AR channel target latency; a value of 0 corresponds to no regulation	RW	12'h000
[15:12]	Reserved	Reserved	RO	-
[11:0]	s#{index}_aw_lat_tgt	Port S#{index} AW channel target latency; a value of 0 corresponds to no regulation	RW	12'h000

4.3.14.12 por_rnd_s0-2_qos_lat_scale

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the QoS target latency scale factor for port S#{index} read and write transactions. This register represents powers of two from the range 2^(-5) to 2^(-12); it is used to match a 16-bit integrator.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA90 + #{index*32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-572: por_rnd_s0-2_qos_lat_scale

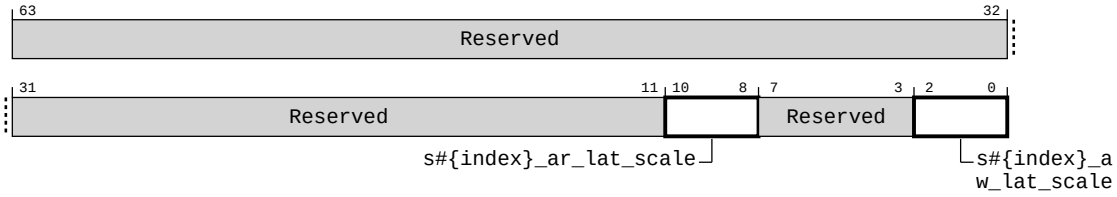


Table 4-588: por_rnd_s0-2_qos_lat_scale attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10:8]	s#{index}_ar_lat_scale	Port S#{index} AR QoS scale factor **3'b000** 2^(-5) **3'b001** 2^(-6) **3'b010** 2^(-7) **3'b011** 2^(-8) **3'b100** 2^(-9) **3'b101** 2^(-10) **3'b110** 2^(-11) **3'b111** 2^(-12)	RW	3'h0
[7:3]	Reserved	Reserved	RO	-
[2:0]	s#{index}_aw_lat_scale	Port S#{index} AW QoS scale factor **3'b000** 2^(-5) **3'b001** 2^(-6) **3'b010** 2^(-7) **3'b011** 2^(-8) **3'b100** 2^(-9) **3'b101** 2^(-10) **3'b110** 2^(-11) **3'b111** 2^(-12)	RW	3'h0

4.3.14.13 por_rnd_s0-2_qos_lat_range

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the minimum and maximum QoS values generated by the QoS latency regulator for port S#{index} read and write transactions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA98 + #{index*32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnd_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-573: por_rnd_s0-2_qos_lat_range

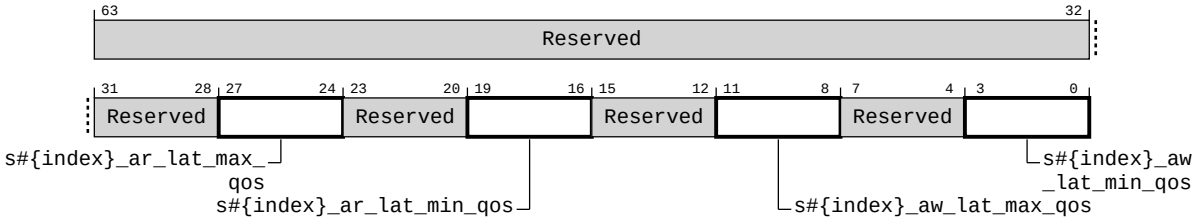


Table 4-589: por_rnd_s0-2_qos_lat_range attributes

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:24]	s#{index}_ar_lat_max_qos	Port S#{index} AR QoS maximum value	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	s#{index}_ar_lat_min_qos	Port S#{index} AR QoS minimum value	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	s#{index}_aw_lat_max_qos	Port S#{index} AW QoS maximum value	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	s#{index}_aw_lat_min_qos	Port S#{index} AW QoS minimum value	RW	4'h0

4.3.14.14 por_rnd_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-574: por_rnd_pmu_event_sel

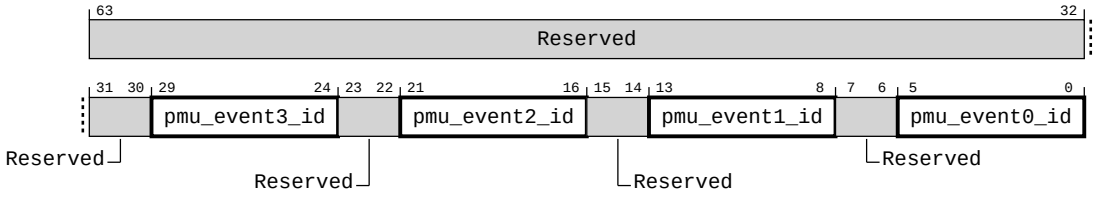


Table 4-590: por_rnd_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	RN-D PMU Event 3 ID; see pmu_event0_id for encodings	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	RN-D PMU Event 2 ID; see pmu_event0_id for encodings	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	RN-D PMU Event 1 ID; see pmu_event0_id for encodings	RW	6'b0
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	RN-D PMU Event 0 ID **6'h00** No event **6'h01** Port S0 RDataBeats **6'h02** Port S1 RDataBeats **6'h03** Port S2 RDataBeats **6'h04** RXDAT flits received **6'h05** TXDAT flits sent **6'h06** Total TXREQ flits sent **6'h07** Retried TXREQ flits sent **6'h08** RRT occupancy count overflow_slice0 **6'h09** WRT occupancy count overflow **6'h0A** Replayed TXREQ flits **6'h0B** WriteCancel sent **6'h0C** Port S0 WDataBeats **6'h0D** Port S1 WDataBeats **6'h0E** Port S2 WDataBeats **6'h0F** RRT allocation **6'h10** WRT allocation **6'h11** PADB occupancy count overflow **6'h12** RPDB occupancy count overflow **6'h13** RRT occupancy count overflow_slice1 **6'h14** RRT occupancy count overflow_slice2 **6'h15** RRT occupancy count overflow_slice3 **6'h16** WRT request throttled **6'h17** RNI backpressure CHI LDB full **6'h18** RRT normal rd req occupancy count overflow_slice0 **6'h19** RRT normal rd req occupancy count overflow_slice1 **6'h1A** RRT normal rd req occupancy count overflow_slice2 **6'h1B** RRT normal rd req occupancy count overflow_slice3 **6'h1C** RRT PCIe RD burst req occupancy count overflow_slice0 **6'h1D** RRT PCIe RD burst req occupancy count overflow_slice1 **6'h1E** RRT PCIe RD burst req occupancy count overflow_slice2 **6'h1F** RRT PCIe RD burst req occupancy count overflow_slice3 **6'h20** RRT PCIe RD burst allocation **6'h21** Compressed AWID ordering **6'h22** Atomic data buffer allocation **6'h23** Atomic data buffer occupancy	RW	6'b0

4.3.14.15 por_rnd_syscoreq_ctl

Functions as the RN-D DVM domain control register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por_rnd_syscoack_status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C00

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-575: por_rnd_syscoreq_ctl

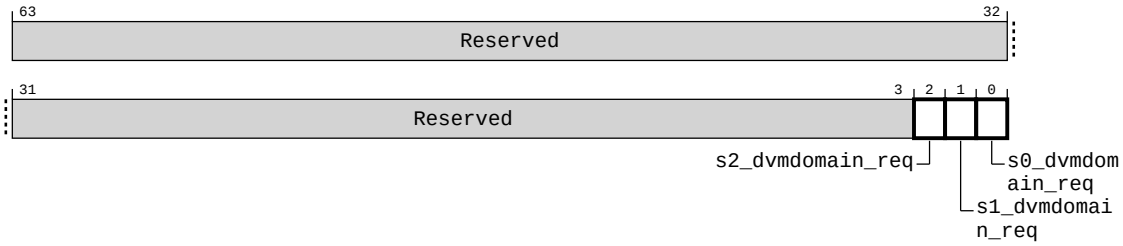


Table 4-591: por_rnd_syscoreq_ctl attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	s2_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for port S2	RW	1'b0
[1]	s1_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for port S1	RW	1'b0
[0]	s0_dvmdomain_req	Controls DVM domain enable (SYSCOREQ) for port S0	RW	1'b0

4.3.14.16 por_rnd_syscoack_status

Functions as the RN-D DVM domain status register. Provides a software alternative to hardware SYSCOREQ/SYSCOACK handshake. Works with por_rnd_syscoreq_ctl.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C08

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-576: por_rnd_syscoack_status

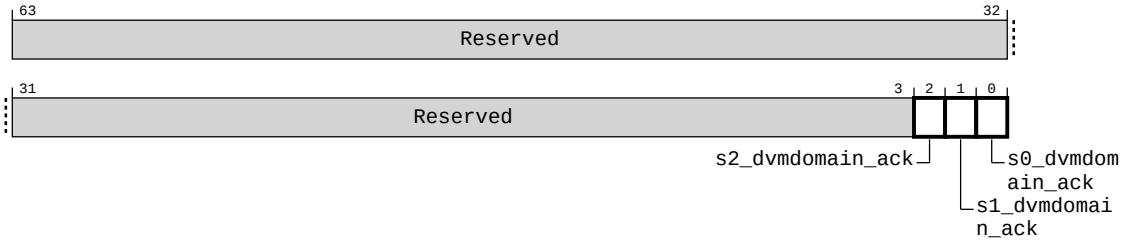


Table 4-592: por_rnd_syscoack_status attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	s2_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for port S2	RO	1'b0
[1]	s1_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for port S1	RO	1'b0
[0]	s0_dvmdomain_ack	Provides DVM domain status (SYSCOACK) for port S0	RO	1'b0

4.3.15 RN-I register descriptions

This section lists the RN-I registers.

4.3.15.1 por_rni_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-577: por_rni_node_info

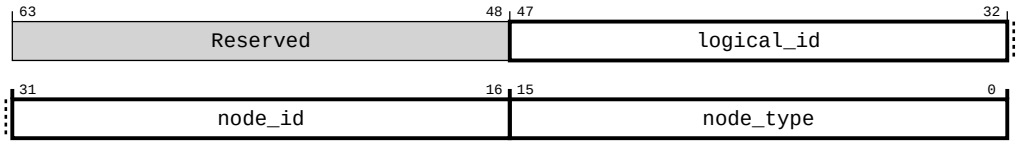


Table 4-593: por_rni_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h000A

4.3.15.2 por_rni_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-578: por_rni_child_info

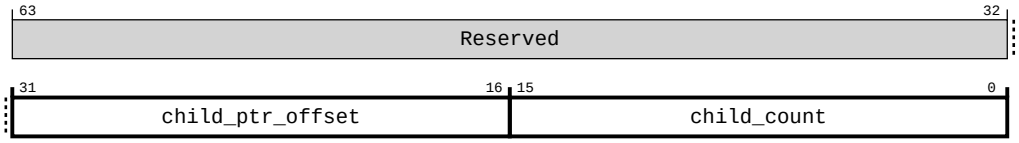


Table 4-594: por_rni_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'h0

4.3.15.3 por_rni_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-579: por_rni_secure_register_groups_override

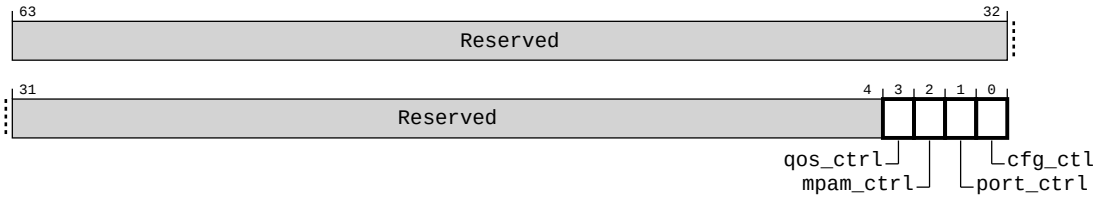


Table 4-595: por_rni_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	qos_ctrl	Allows Non-secure access to Secure QoS control registers	RW	1'b0
[2]	mpam_ctrl	Allows Non-secure access to Secure AXI port MPAM override register	RW	1'b0
[1]	port_ctrl	Allows Non-secure access to Secure AXI port control registers	RW	1'b0
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register	RW	1'b0

4.3.15.4 por_rni_unit_info

Provides component identification information for RN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-580: por_rni_unit_info

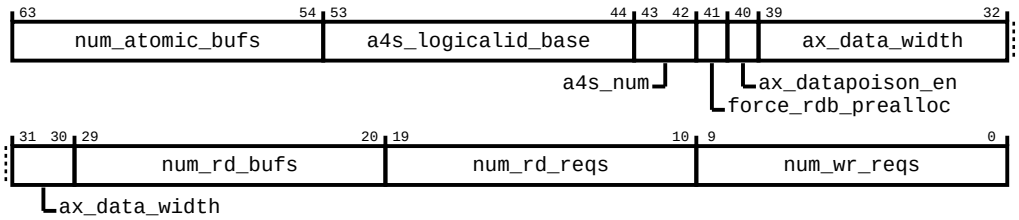


Table 4-596: por_rni_unit_info attributes

Bits	Name	Description	Type	Reset
[63:54]	num_atomic_bufs	Number of atomic data buffers	RO	Configuration dependent
[53:44]	a4s_logicalid_base	AXI4Stream interfaces logical ID base	RO	Configuration dependent
[43:42]	a4s_num	Number of AXI4Stream interfaces present	RO	Configuration dependent
[41]	force_rdb_prealloc	Force read data buffer preallocation **1'b1** Yes **1'b0** No	RO	Configuration dependent
[40]	ax_datapoison_en	Data Poison enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[39:30]	ax_data_width	AXI interface data width in bits	RO	Configuration dependent
[29:20]	num_rd_bufs	Number of read data buffers	RO	Configuration dependent
[19:10]	num_rd_reqs	Number of outstanding read requests	RO	Configuration dependent
[9:0]	num_wr_reqs	Number of outstanding write requests	RO	Configuration dependent

4.3.15.5 por_rni_unit_info2

Provides additonal component identification information for RN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-581: por_rni_unit_info2

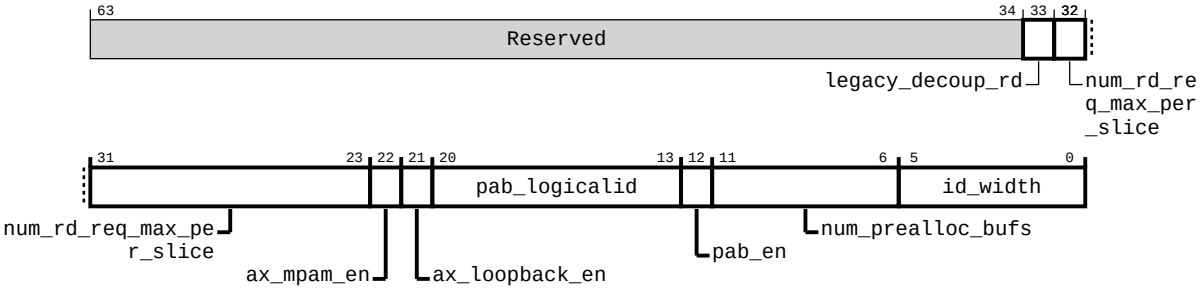


Table 4-597: por_rni_unit_info2 attributes

Bits	Name	Description	Type	Reset
[63:34]	Reserved	Reserved	RO	-
[33]	legacy_decoup_rd	Legacy decoupled read mode, no read burst propagation	RO	Configuration dependent
[32:23]	num_rd_req_max_per_slice	Number of read request entires per slice	RO	Configuration dependent
[22]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[21]	ax_loopback_en	LoopBack enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[20:13]	pab_logicalid	PUB AUB bridge Logical ID	RO	Configuration dependent
[12]	pab_en	PUB AUB bridge enable **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[11:6]	num_prealloc_bufs	Number of Pre-allocated Read Data Buffers	RO	Configuration dependent
[5:0]	id_width	AXI ID width for ACE-Lite subordinate ports	RO	Configuration dependent

4.3.15.6 por_rni_cfg_ctl

Functions as the configuration control register. Specifies the current mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA00

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-582: por_rni_cfg_ctl

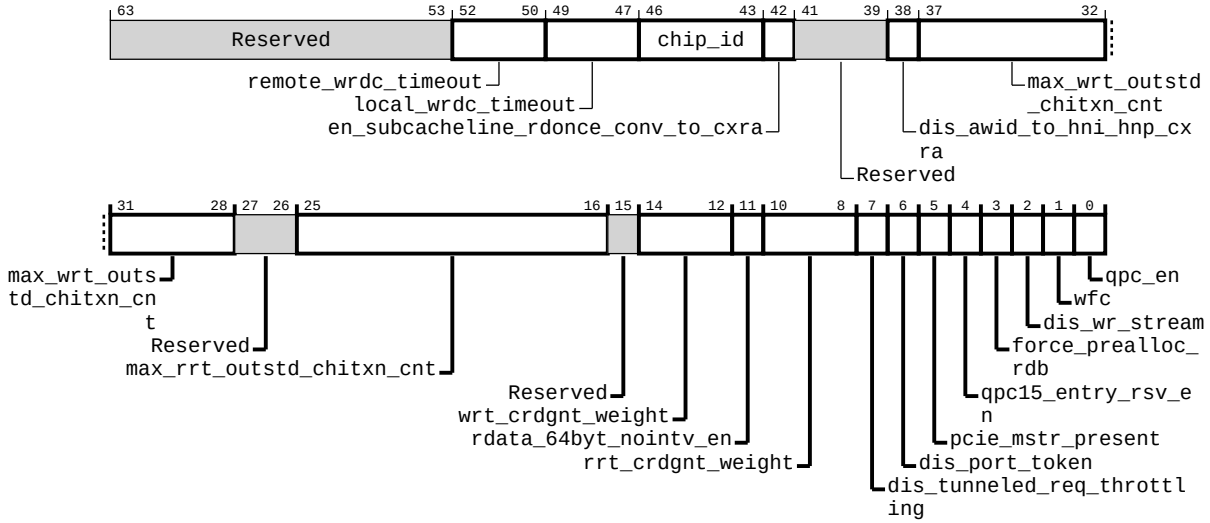


Table 4-598: por_rni_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:53]	Reserved	Reserved	RO	-
[52:50]	remote_wrdc_timeout	Configurable write data cancel timeout value for remote traffic. **3'b000** ~2000 cycles **3'b001** ~4000 cycles **3'b010** ~8000 cylces **3'b011** ~16000 cycles (default) **3'b100** ~32000 cycles (all other values rsvd)	RW	3'b011
[49:47]	local_wrdc_timeout	Configurable write data cancel timeout value for local traffic. **3'b000** ~2000 cycles **3'b001** ~4000 cycles **3'b010** ~8000 cylces **3'b011** ~16000 cycles (default) **3'b100** ~32000 cycles (all other values rsvd)	RW	3'b010
[46:43]	chip_id	Configurable ChipID for this RNX instance. Must be correctly set for proper handling of remote traffic to HNI/HNP. Only supports values 0..3. Two MSB's is reserved.	RW	4'b0000
[42]	en_subcacheline_rdonce_conv_to_cxra	If set, enables the conversion of sub-cacheline RdOnce to RdNoSnp for CXRA targets	RW	1'b0
[41:39]	Reserved	Reserved	RO	-
[38]	dis_awid_to_hni_hnp_cxra	If set, disables compressed AWID to HNI, HNP and CXRA, also disables compressed AWID based ordering. Set this bit if uniq-ID write performance is needed.	RW	1'b0
[37:28]	max_wrt_outstd_chitxn_cnt	Maximum number of outstanding writes allowed on CHI-side	RW	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	max_rrt_outstd_chitxn_cnt	Maximum number of outstanding reads allowed on CHI-side	RW	Configuration dependent
[15]	Reserved	Reserved	RO	-
[14:12]	wrt_crdgnt_weight	Determines weight of credit grant allocated to retried writes in presence of pending retried reads	RW	3'b001
[11]	rdata_64byt_nointv_en	Enables no interleaving property on normal memory read data within 64B granule when set	RW	1'b1
[10:8]	rrt_crdgnt_weight	Determines weight of credit grant allocated to retried reads in presence of pending retried writes	RW	3'b100
[7]	dis_tunneled_req_throttling	Disables retry based throttling of tunneled write requests	RW	1'b0
[6]	dis_port_token	If set, disables per port reservation in the tracker(rd and wr)	RW	1'b1
[5]	pcie_mstr_present	Indicates PCIe manager is present; must be set if PCIe manager is present upstream of RN-I or RN-D	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	qpc15_entry_rsv_en	Enables QPC15 entry reservation **1'b1** Reserves tracker entry for QoS15 requests **1'b0** Does not reserve tracker entry for QoS15 requests NOTE: Only valid and applicable when por_rnd_qpc_en is set	RW	1'b0
[3]	force_prealloc_rdb	When set, all reads from the RN-I are sent with a preallocated read data buffer	RW	Configuration dependent
[2]	dis_wr_stream	Disables streaming of ordered writes when set	RW	1'b0
[1]	wfc	When set, enables waiting for completion (COMP) before dispatching dependent transaction (TXN)	RW	1'b0
[0]	qpc_en	When set, enables QPC-based scheduling using two QoS priority classes (QoS15 and non-QoS15)	RW	1'b1

4.3.15.7 por_rni_aux_ctl

Functions as the auxiliary control register for RN-I.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-583: por_rni_aux_ctl

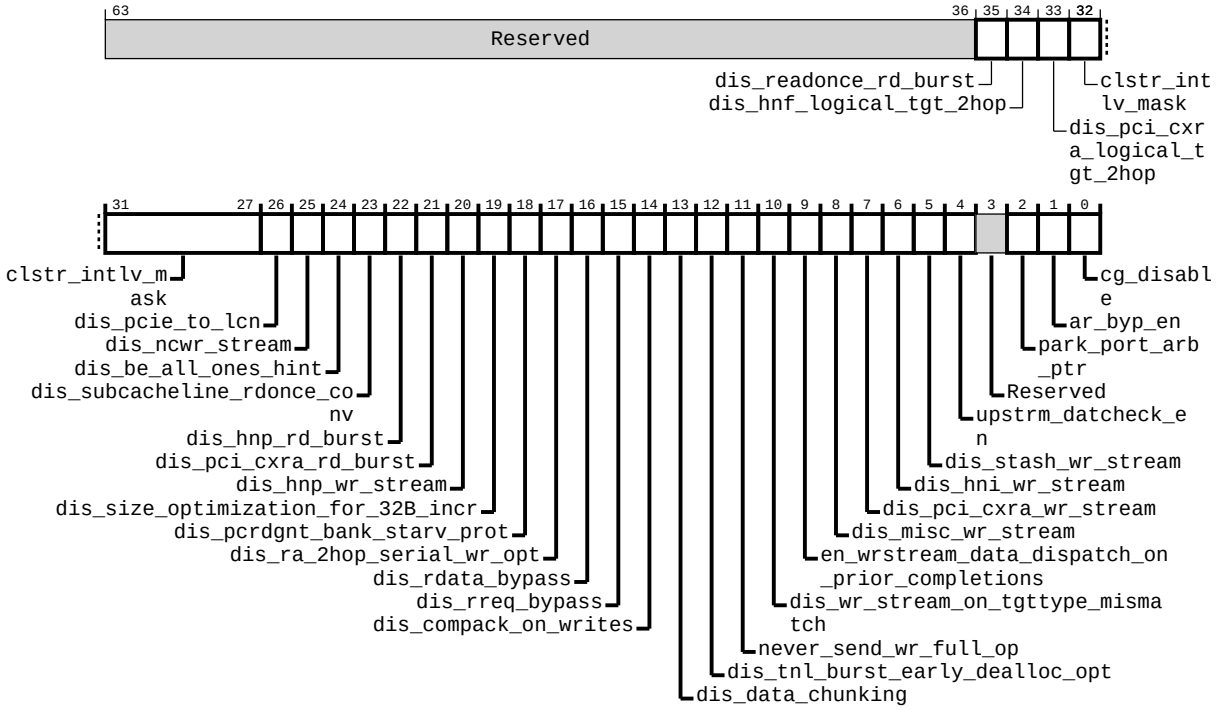


Table 4-599: por_rni_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[39]	sys_dis_data_interleaving,	System optimized disable read DATA interleaving for all ports. Disables all read data interleaving, including atomic read data being returned for all AXI ports. Read burst preservation is enabled similar to normal mode, but this requires certain system level restrictions: 1. Cannot set SYS_DIS_DATA_INTERLEAVING for multi-chip systems. Support for remote HN-P is a future feature. 2. When setting SYS_DIS_DATA_INTERLEAVING for an RN-I/RN-D it is required for that RN-I/RN-D to target only one HN-P with read bursts. It is also required for the path to a single remote HN-P, when remote support is available, to only go through a single CCG and for the ccg_rni to also have SYS_DIS_DATA_INTERLEAVING set. 3. The AXI subordinate downstream of HN-P must not interleave read burst data. 4. Must have NUM_RD_REQ_PARAM==NUM_RD_BUF_PARAM and NUM_RD_REQ_PRAM<=256, otherwise this bit has no effect. 5. The sum of the maximum number of beats of a cracked burst per each port must fit wholly within an arslice - breaking this rule will result in a hang. Must comprehend DIS_PORT_TOKEN AND QPC15_ENTRY_RSV settings, which will limit number of available entries	RW	1'b0
[38:36]	Reserved	Reserved	RO	-
[35]	dis_readonce_rd_burst	If set, disables read burst for ReadOnce from AXI.	RW	1'b0
[34]	dis_hnf_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical target is pci_cxra and logical target is hnf, otherwise may tunneling/2hop to RA if interleaving granularity settings allow.	RW	1'b0
[33]	dis_pci_cxra_logical_tgt_2hop	If set, disables tunneling/2hop for case where physical and logical target is pci_cxra, otherwise tunneling/2hop to RA.	RW	1'b0
[32:27]	clstr_intlv_mask	Encoded static mask for max interleave granularity supported. When this setting is less than or equal to rnsam's programmed interleave granularity for a write to pci_cxra, tunneling/2hop flow will be used. **6'b111111** 64B **6'b111110** 128B **6'b111100** 256B **6'b111000** 512B **6'b110000** 1024B **6'b100000** 2048B **6'b000000** 4096B **Others** Reserved	RW	6'b000000
[26]	dis_pcie_to_lcn	If set, all pcie traffic sent directly to HNF/CCG, bypasses LCN. Only has effect when pcie_mstr_present	RW	1'b1
[25]	dis_ncwr_stream	Disables streaming of ordered non-cacheable writes when set	RW	1'b0

Bits	Name	Description	Type	Reset
[24]	dis_be_all_ones_hint	If set, disables hint to HNF which signals all BE=1's on writes	RW	1'b0
[23]	dis_subcacheline_rdonce_conv	If set, disables the conversion of sub-cacheline RdOnce to RdNoSnp across all targets	RW	1'b0
[22]	dis_hnp_rd_burst	If set, disables read burst to HNP on CHI request flits . Read burst on CHI is supported only in non-decoupled RDB configuration.	RW	1'b0
[21]	dis_pci_cxra_rd_burst	If set, disables read burst to PCI-CXRA on CHI request flits . Read burst on CHI is supported only in non-decoupled RDB configuration.	RW	1'b0
[20]	dis_hnp_wr_stream	Disables streaming of ordered writes to HNP when set	RW	1'b0
[19]	dis_size_optimization_for_32B_incr	If set, disables the size related optimization for a 32B INCR burst (rh-2512). Only applies to writes.	RW	1'b0
[18]	dis_pcrdgnt_bank_starv_prot	If set, disables across arslice starvation protection	RW	1'b0
[17]	dis_ra_2hop_serial_wr_opt	If set, disables 2 hop indication to ra for serilized writes; will indicate 3 hop	RW	1'b0
[16]	dis_rdata_bypass	If set, disables read data bypass path	RW	1'b0
[15]	dis_rreq_bypass	If set, disables read request bypass path	RW	1'b0
[14]	dis_compack_on_writes	If set, disables comp_ack on streaming writes. WrData is used for ordering writes	RW	1'b1
[13]	dis_data_chunking	If set, disables the data chunking feature	RW	1'b0
[12]	dis_tnl_burst_early_dealloc_opt	If set, disables the optimization related to early deallocation of tunnelled writes for intermediate txns of burst	RW	1'b0
[11]	never_send_wr_full_op	If set, RNI will never send WR FULL op. All write ops will be of PTL type	RW	1'b0
[10]	dis_wr_stream_on_tgttype_mismatch	If set, serializes first write when moving from one tgttype to another	RW	1'b0
[9]	en_wrstream_data_dispatch_on_prior_completions	If set, data dispatch for streaming writes waits for completion of all older writes	RW	1'b0
[8]	dis_misc_wr_stream	Disables streaming of ordered writes with following attributes when set : Device memory or EWA=0 or Excl=1	RW	1'b0
[7]	dis_pci_cxra_wr_stream	Disables streaming of ordered writes to PCI-CXRA when set	RW	1'b0
[6]	dis_hni_wr_stream	Disables streaming of ordered writes to HNI when set	RW	1'b0
[5]	dis_stash_wr_stream	Disables streaming of ordered WrUniqStash when set	RW	1'b0
[4]	upstrm_datcheck_en	Upstream supports Datacheck	RW	Configuration dependent
[3]	Reserved	Reserved	RO	-
[2]	park_port_arb_ptr	Parks the AXI port arbitration pointer for Burst	RW	1'b0
[1]	ar_byp_en	AR bypass enable; enables bypass path in the AR pipeline	RW	1'b1
[0]	cg_disable	Disables clock gating when set	RW	1'b0

4.3.15.8 por_rni_s0-2_port_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface settings.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA10 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.port_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-584: por_rni_s0-2_port_control

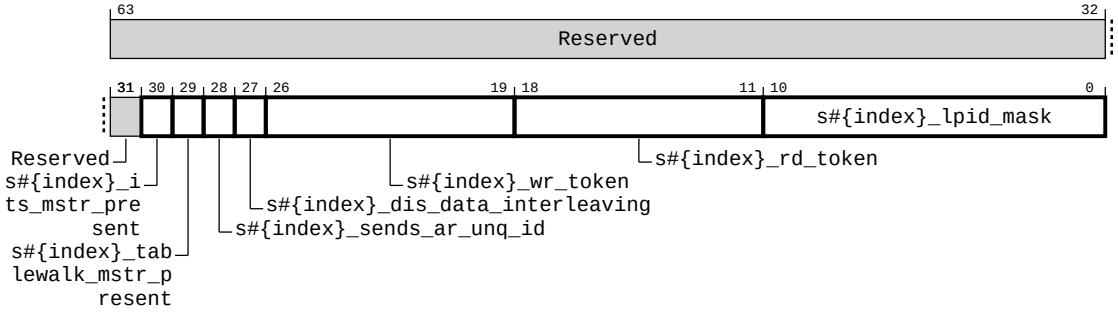


Table 4-600: por_rni_s0-2_port_control attributes

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[30]	s#{index}_its_mstr_present	Must be set if translation table walk manager present such as TCU or GIC for non-PCIE case. This affects RND AW channel only.	RW	1'b0
[29]	s#{index}_tablewalk_mstr_present	Must be set if translation table walk manager present such as TCU or GIC. This affects RND AR channel only.	RW	1'b0
[28]	s#{index}_sends_ar_unq_id	If set, indicates AR transactions on Port#{index} are always Unique ID. This bit for a port must be set to 1 to enable Read Burst on the CHI side of RND.	RW	1'b0
[27]	s#{index}_dis_data_interleaving	If set, disables read DATA interleaving on RDATAS#{index} channel. This applies only to RDATA generated as a response to requests on AR channel . This does not apply to RDATA generated as a response to Atomic request on AW channel. I.e. RDATA of an Atomic op, on AW channel, may interleave with RDATA of an AR channel request	RW	1'b0
[26:19]	s#{index}_wr_token	Port S#{index} reserved token count for AW channel This must be less than the number of Wr requests(RNID_NUM_XRT_REQ) on AW achnnel	RW	8'b0000_0000
[18:11]	s#{index}_rd_token	Port S#{index} reserved token count for AR channel per slice This should be less than the number of Rd requests(RNID_NUM_XRT_SLICE_REQ) per slice on AR achnnel	RW	8'b0000_0000
[10:0]	s#{index}_lpid_mask	Port S#{index} LPID mask LPID[0]: Equal to the result of UnaryOR of BitwiseAND of LPID mask and AXID (LPID[0] = |(AXID & mask)); specifies which AXID bit is reflected in the LSB of LPID LPID[2:1]: Equal to port ID[1:0]; the MSB of LPID contains port ID	RW	11'b000_0000_0000

4.3.15.9 por_rni_s0-2_mpam_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls port S#{index} AXI/ACE subordinate interface MPAM override values

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA28 + #{index*8}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.mpam_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-585: por_rni_s0-2_mpam_control

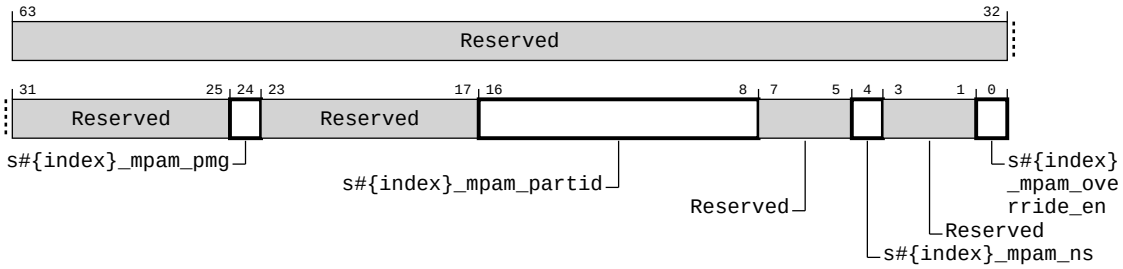


Table 4-601: por_rni_s0-2_mpam_control attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	s#{index}_mpam_pmg	Port S#{index} MPAM_PMG value	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	s#{index}_mpam_partid	Port S#{index} MPAM_PARTID value	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	s#{index}_mpam_ns	Port S#{index} MPAM_NS value	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	s#{index}_mpam_override_en	Port S#{index} MPAM override en When set, MPAM value on CHI side is driven from MPAM override value in this register. Note that when RNID_AXMPAM_EN_PARAM is set to 0, MPAM override value is always used irrespective of this bit value	RW	1'b0

4.3.15.10 por_rni_s0-2_qos_control

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS settings for port S#{index} AXI/ACE subordinate interface.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
16'hA80 + #{index*32}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-586: por_rni_s0-2_qos_control

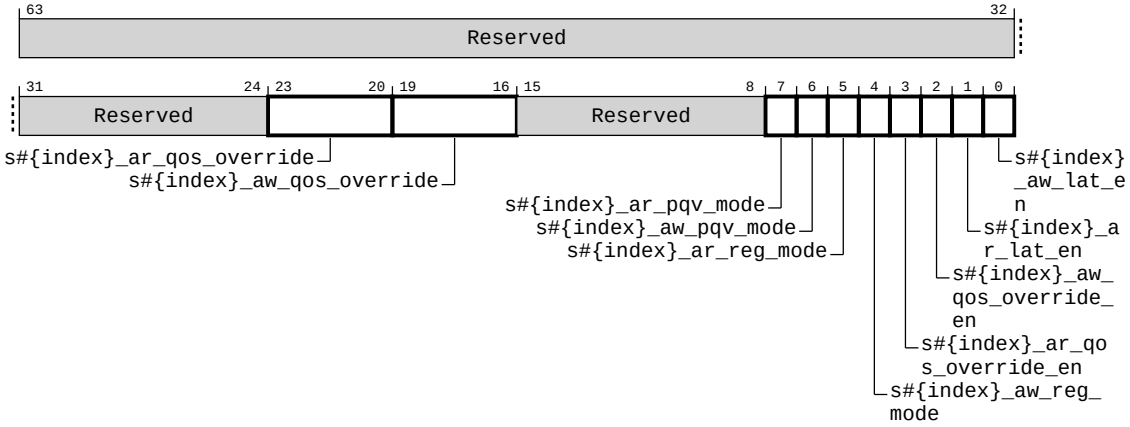


Table 4-602: por_rni_s0-2_qos_control attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	s#{index}_ar_qos_override	AR QoS override value for port S#{index}	RW	4'b0000
[19:16]	s#{index}_aw_qos_override	AW QoS override value for port S#{index}	RW	4'b0000
[15:8]	Reserved	Reserved	RO	-
[7]	s#{index}_ar_pqv_mode	Configures the QoS regulator mode for read transactions during period mode **1'b0** Normal mode; QoS value is stable when the manager is idle **1'b1** Quiesce high mode; QoS value tends to the maximum value when the manager is idle	RW	1'b0
[6]	s#{index}_aw_pqv_mode	Configures the QoS regulator mode for write transactions during period mode **1'b0** Normal mode; QoS value is stable when the manager is idle **1'b1** Quiesce high mode; QoS value tends to the maximum value when the manager is idle	RW	1'b0

Bits	Name	Description	Type	Reset
[5]	s#{index}_ar_reg_mode	Configures the QoS regulator mode for read transactions **1'b0** Latency mode **1'b1** Period mode; used for bandwidth regulation	RW	1'b0
[4]	s#{index}_aw_reg_mode	Configures the QoS regulator mode for write transactions **1'b0** Latency mode **1'b1** Period mode; used for bandwidth regulation	RW	1'b0
[3]	s#{index}_ar_qos_override_en	Enables port S#{index} AR QoS override; when set, allows QoS value on inbound AR transactions to be overridden	RW	1'b0
[2]	s#{index}_aw_qos_override_en	Enables port S#{index} AW QoS override; when set, allows QoS value on inbound AW transactions to be overridden	RW	1'b0
[1]	s#{index}_ar_lat_en	Enables port S#{index} AR QoS regulation when set	RW	1'b0
[0]	s#{index}_aw_lat_en	Enables port S#{index} AW QoS regulation when set	RW	1'b0

4.3.15.11 por_rni_s0-2_qos_lat_tgt

There are 3 iterations of this register. The index ranges from 0 to 2. Controls QoS target latency (in cycles) for regulations of port S#{index} read and write transactions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA88 + #{index*32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-587: por_rni_s0-2_qos_lat_tgt

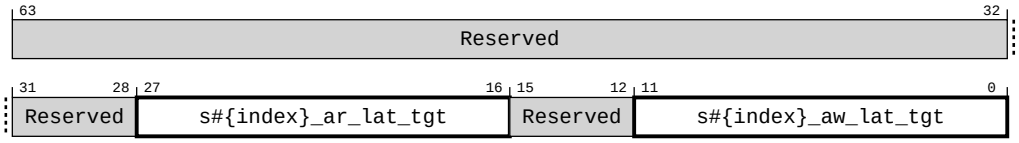


Table 4-603: por_rni_s0-2_qos_lat_tgt attributes

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:16]	s#{index}_ar_lat_tgt	Port S#{index} AR channel target latency; a value of 0 corresponds to no regulation	RW	12'h000
[15:12]	Reserved	Reserved	RO	-
[11:0]	s#{index}_aw_lat_tgt	Port S#{index} AW channel target latency; a value of 0 corresponds to no regulation	RW	12'h000

4.3.15.12 por_rni_s0-2_qos_lat_scale

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the QoS target latency scale factor for port S#{index} read and write transactions. This register represents powers of two from the range 2^(-5) to 2^(-12); it is used to match a 16-bit integrator.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA90 + #{index*32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-588: por_rni_s0-2_qos_lat_scale

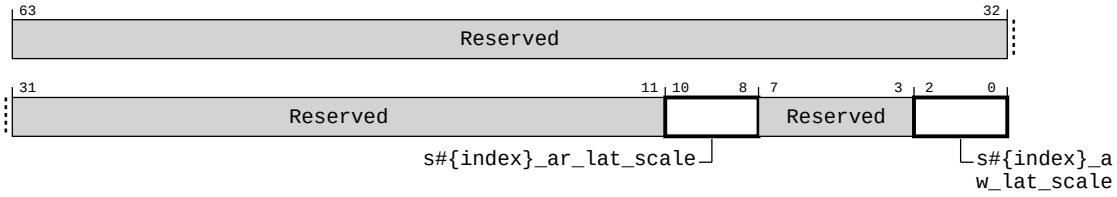


Table 4-604: por_rni_s0-2_qos_lat_scale attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10:8]	s#{index}_ar_lat_scale	Port S#{index} AR QoS scale factor **3'b000** 2^(-5) **3'b001** 2^(-6) **3'b010** 2^(-7) **3'b011** 2^(-8) **3'b100** 2^(-9) **3'b101** 2^(-10) **3'b110** 2^(-11) **3'b111** 2^(-12)	RW	3'h0
[7:3]	Reserved	Reserved	RO	-
[2:0]	s#{index}_aw_lat_scale	Port S#{index} AW QoS scale factor **3'b000** 2^(-5) **3'b001** 2^(-6) **3'b010** 2^(-7) **3'b011** 2^(-8) **3'b100** 2^(-9) **3'b101** 2^(-10) **3'b110** 2^(-11) **3'b111** 2^(-12)	RW	3'h0

4.3.15.13 por_rni_s0-2_qos_lat_range

There are 3 iterations of this register. The index ranges from 0 to 2. Controls the minimum and maximum QoS values generated by the QoS latency regulator for port S#{index} read and write transactions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA98 + #{index*32}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rni_secure_register_groups_override.qos_ctrl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-589: por_rni_s0-2_qos_lat_range

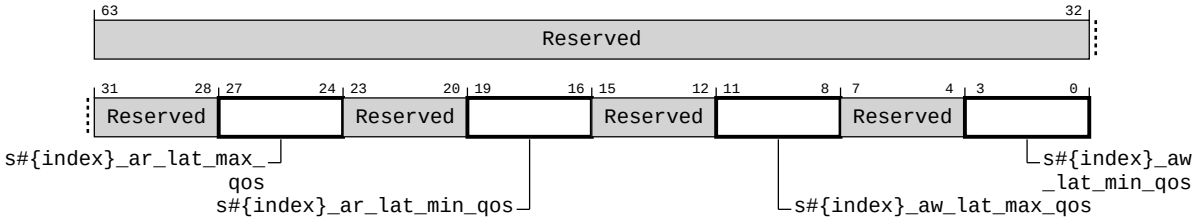


Table 4-605: por_rni_s0-2_qos_lat_range attributes

Bits	Name	Description	Type	Reset
[63:28]	Reserved	Reserved	RO	-
[27:24]	s#{index}_ar_lat_max_qos	Port S#{index} AR QoS maximum value	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	s#{index}_ar_lat_min_qos	Port S#{index} AR QoS minimum value	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	s#{index}_aw_lat_max_qos	Port S#{index} AW QoS maximum value	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	s#{index}_aw_lat_min_qos	Port S#{index} AW QoS minimum value	RW	4'h0

4.3.15.14 por_rni_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-590: por_rni_pmu_event_sel

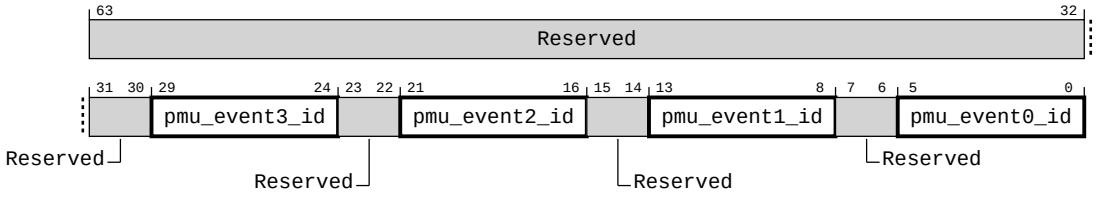


Table 4-606: por_rni_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	RN-I PMU Event 3 ID; see pmu_event0_id for encodings	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	RN-I PMU Event 2 ID; see pmu_event0_id for encodings	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	RN-I PMU Event 1 ID; see pmu_event0_id for encodings	RW	6'b0
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	RN-I PMU Event 0 ID **6'h00** No event **6'h01** Port S0 RDataBeats **6'h02** Port S1 RDataBeats **6'h03** Port S2 RDataBeats **6'h04** RXDAT flits received **6'h05** TXDAT flits sent **6'h06** Total TXREQ flits sent **6'h07** Retried TXREQ flits sent **6'h08** RRT occupancy count overflow_slice0 **6'h09** WRT occupancy count overflow **6'h0A** Replayed TXREQ flits **6'h0B** WriteCancel sent **6'h0C** Port S0 WDataBeats **6'h0D** Port S1 WDataBeats **6'h0E** Port S2 WDataBeats **6'h0F** RRT allocation **6'h10** WRT allocation **6'h11** PADB occupancy count overflow **6'h12** RPDB occupancy count overflow **6'h13** RRT occupancy count overflow_slice1 **6'h14** RRT occupancy count overflow_slice2 **6'h15** RRT occupancy count overflow_slice3 **6'h16** WRT request throttled **6'h17** RNI backpressure CHI LDB full **6'h18** RRT normal rd req occupancy count overflow_slice0 **6'h19** RRT normal rd req occupancy count overflow_slice1 **6'h1A** RRT normal rd req occupancy count overflow_slice2 **6'h1B** RRT normal rd req occupancy count overflow_slice3 **6'h1C** RRT PCIe RD burst req occupancy count overflow_slice0 **6'h1D** RRT PCIe RD burst req occupancy count overflow_slice1 **6'h1E** RRT PCIe RD burst req occupancy count overflow_slice2 **6'h1F** RRT PCIe RD burst req occupancy count overflow_slice3 **6'h20** RRT PCIe RD burst allocation **6'h21** Compressed AWID ordering **6'h22** Atomic data buffer allocation **6'h23** Atomic data buffer occupancy	RW	6'b0

4.3.16 RN SAM register descriptions

This section lists the RN SAM registers.

4.3.16.1 por_rnsam_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-591: por_rnsam_node_info

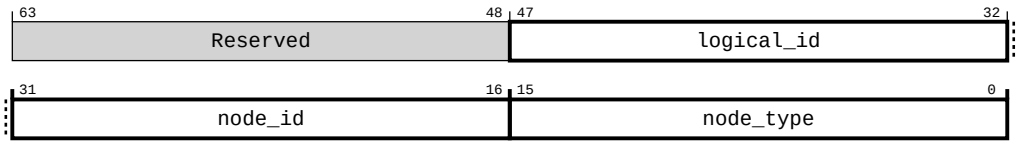


Table 4-607: por_rnsam_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID NOTE: RN SAM logical ID is always set to 16'b0.	RO	16'h0
[31:16]	node_id	Component node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h000F

4.3.16.2 por_rnsam_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h80

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-592: por_rnsam_child_info

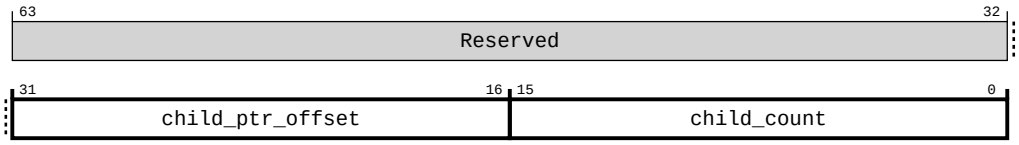


Table 4-608: por_rnsam_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.16.3 por_rnsam_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h980

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-593: por_rnsam_secure_register_groups_override

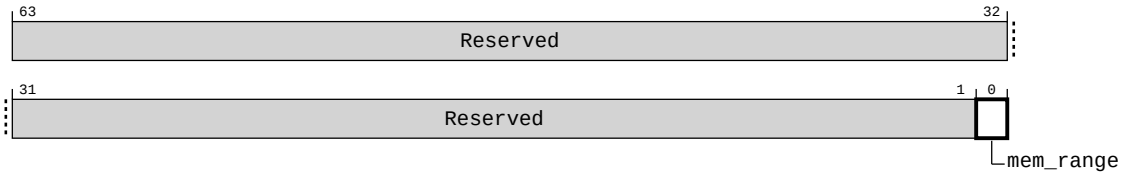


Table 4-609: por_rnsam_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	mem_range	Allows Non-secure access to Secure mem_ranges registers	RW	1'b0

4.3.16.4 por_rnsam_unit_info

Provides component identification information for RN SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-594: por_rnsam_unit_info

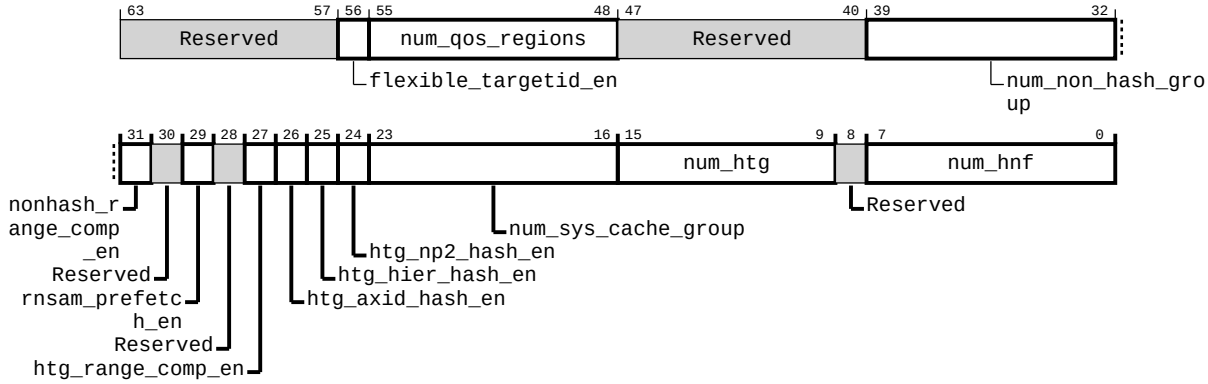


Table 4-610: por_rnsam_unit_info attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56]	flexible_targetid_en	flexible target enable to preserve backward compatability	RO	Configuration dependent
[55:48]	num_qos_regions	Number of QOS regions	RO	Configuration dependent
[47:40]	Reserved	Reserved	RO	-
[39:32]	num_non_hash_group	Number of non-hashed groups supported	RO	Configuration dependent
[31]	nonhash_range_comp_en	Define start and end address for each HTG region	RO	Configuration dependent
[30]	Reserved	Reserved	RO	-
[29]	rnsam_prefetch_en	RNSAM prefetch enabled	RO	Configuration dependent
[28]	Reserved	Reserved	RO	-
[27]	htg_range_comp_en	Define start and end address for each HTG region	RO	Configuration dependent
[26]	htg_axid_hash_en	Enable AXID based hashing scheme	RO	Configuration dependent
[25]	htg_hier_hash_en	Enable Hierarchical hashing scheme	RO	Configuration dependent
[24]	htg_np2_hash_en	Enable non-power of two hash scheme	RO	Configuration dependent
[23:16]	num_sys_cache_group	Number of system cache groups supported This register field value has deprecated its reset value is always 0	RO	0
[15:9]	num_htg	Number of Hashed target groups	RO	Configuration dependent
[8]	Reserved	Reserved	RO	-
[7:0]	num_hnf	Number of hashed targets supported	RO	Configuration dependent

4.3.16.5 por_rnsam_unit_info1

Provides component identification information for RN SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h908`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-595: por_rnsam_unit_info1

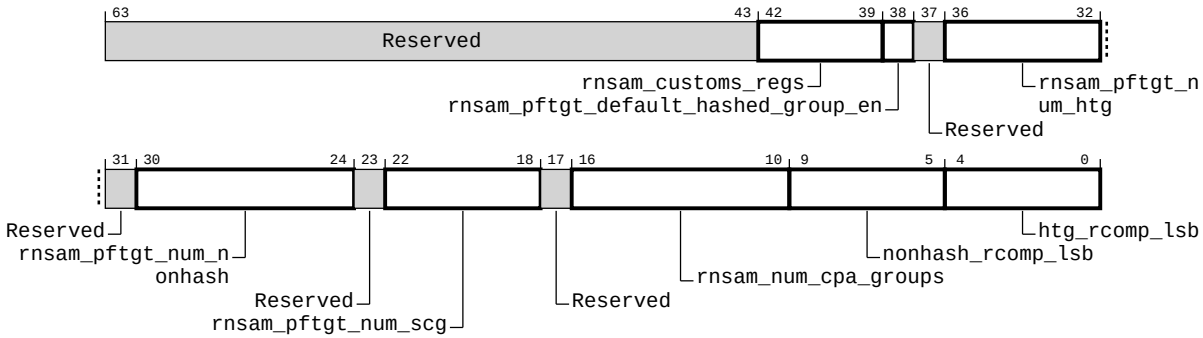


Table 4-611: por_rnsam_unit_info1 attributes

Bits	Name	Description	Type	Reset
[63:43]	Reserved	Reserved	RO	-
[42:39]	rnsam_customs_regs	Number of customer specific registers for customer implemented logic	RO	Configuration dependent
[38]	rnsam_pftgt_default_hashed_group_en	Enable default hashed group for prefetch transactions. To support backward compatible, set this parameter	RO	Configuration dependent
[37]	Reserved	Reserved	RO	-
[36:32]	rnsam_pftgt_num_htg	Number of prefetch HTG regions supported per System Cache Group by the RNSAM	RO	Configuration dependent
[31]	Reserved	Reserved	RO	-
[30:24]	rnsam_pftgt_num_nonhash	Number of prefetch non-hashed regions supported per System Cache Group by the RNSAM	RO	Configuration dependent
[23]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[22:18]	rnsam_pftgt_num_scg	Number of system cache groups enabled for prefetch targets	RO	Configuration dependent
[17]	Reserved	Reserved	RO	-
[16:10]	rnsam_num_cpa_groups	Number of CPA groups	RO	Configuration dependent
[9:5]	nonhash_rcomp_lsb	NONHASH RCOMP LSB bit position defining minimum region size	RO	Configuration dependent
[4:0]	htg_rcomp_lsb	HTG RCOMP LSB bit position defining minimum region size	RO	Configuration dependent

4.3.16.6 non_hash_mem_region_reg0-63

There are 64 iterations of this register. The index ranges from 0 to 63. Configures non-hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-23) : 16'hC00 + #{8 * index}
index(24-63) : 16'h2000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-596: non_hash_mem_region_reg0-63

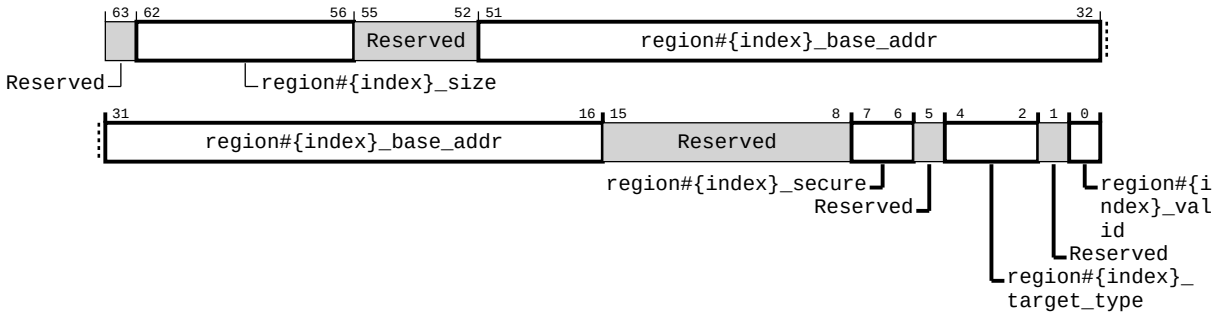


Table 4-612: non_hash_mem_region_reg0-63 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	region#{index}_size	Memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'h0
[55:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_NONHASH_RCOMP_LSB_PARAM	RW	36'h0
[15:8]	Reserved	Reserved	RO	-
[7:6]	region#{index}_secure	Indicates Secure type **2'b00** Trusted device. **2'b01** Trusted device attached memory range only **2'b10** Untrusted device **2'b11** Reserved	RW	2'b10
[5]	Reserved	Reserved	RO	-
[4:2]	region#{index}_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000

Bits	Name	Description	Type	Reset
[1]	Reserved	Reserved	RO	-
[0]	region#{index}_valid	Memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.7 non_hash_mem_region_cfg2_reg0-63

There are 64 iterations of this register. The index ranges from 0 to 63. Configures non-hashed memory region end address

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-23) : 16'hCC0 + #{8 * index}
index(24-151) : 16'h2400 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-597: non_hash_mem_region_cfg2_reg0-63

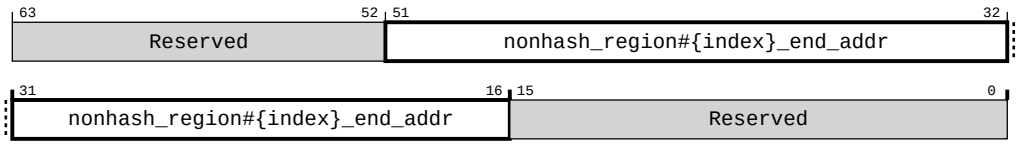


Table 4-613: non_hash_mem_region_cfg2_reg0-63 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	nonhash_region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAM_NONHASH_RCOMP_LSB_PARAM	RW	36'h0
[15:0]	Reserved	Reserved	RO	-

4.3.16.8 non_hash_tgt_nodeid0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures non-hashed target node IDs #{4*index} to #{4*index + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-15) : 16'hD80 + #{8 * index}
index(16-47) : 16'h2800 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-598: non_hash_tgt_nodeid0-15

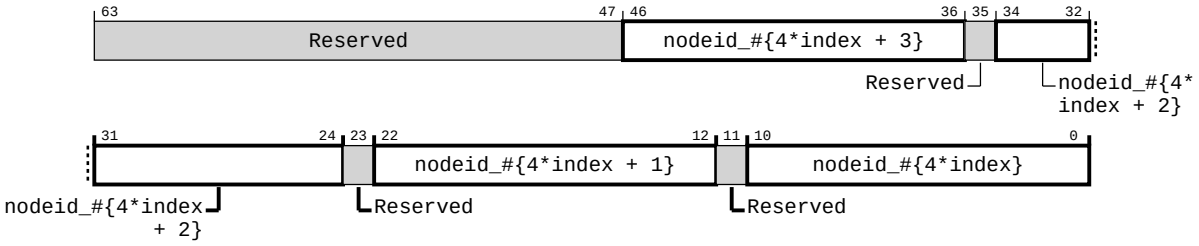


Table 4-614: non_hash_tgt_nodeid0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_#{4*index + 3}	Non-hashed target node ID #{4*index + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_#{4*index + 2}	Non-hashed target node ID #{4*index + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_#{4*index + 1}	Non-hashed target node ID #{4*index + 1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_#{4*index}	Non-hashed target node ID #{4*index}	RW	11'b00000000000

4.3.16.9 cml_port_aggr_mode_ctrl_reg

Configures the CCIX port aggregation modes for all non-hashed memory regions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h11A0`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-599: cml_port_aggr_mode_ctrl_reg

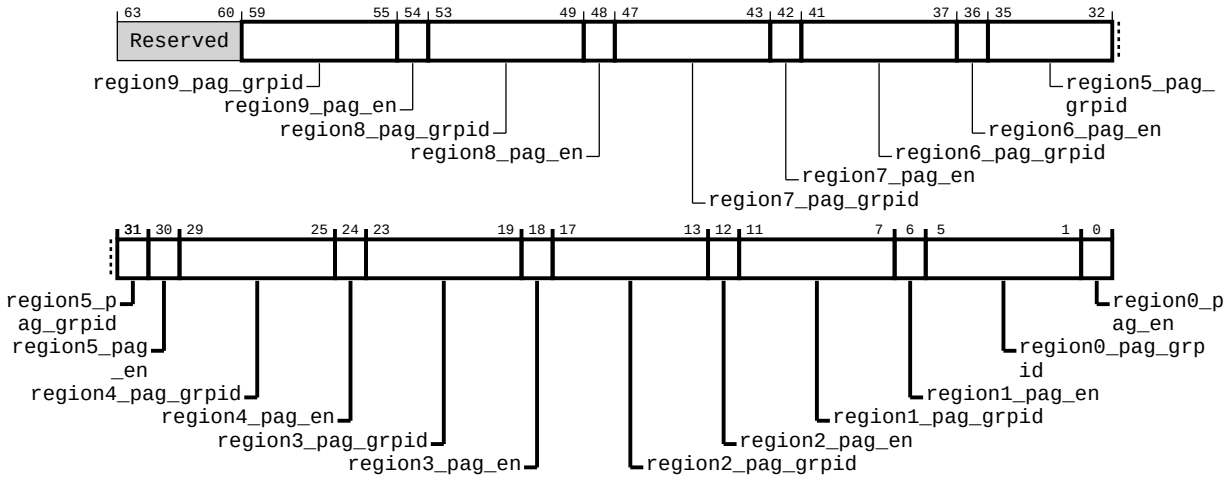


Table 4-615: cml_port_aggr_mode_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:55]	region9_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[54]	region9_pag_en	Enables the CPA mode for non-hashed memory region 9	RW	1'b0
[53:49]	region8_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[48]	region8_pag_en	Enables the CPA mode for non-hashed memory region 8	RW	1'b0
[47:43]	region7_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[42]	region7_pag_en	Enables the CPA mode for non-hashed memory region 7	RW	1'b0
[41:37]	region6_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[36]	region6_pag_en	Enables the CPA mode for non-hashed memory region 6	RW	1'b0
[35:31]	region5_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[30]	region5_pag_en	Enables the CPA mode for non-hashed memory region 5	RW	1'b0
[29:25]	region4_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[24]	region4_pag_en	Enables the CPA mode for non-hashed memory region 4	RW	1'b0
[23:19]	region3_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[18]	region3_pag_en	Enables the CPA mode for non-hashed memory region 3	RW	1'b0
[17:13]	region2_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[12]	region2_pag_en	Enables the CPA mode for non-hashed memory region 2	RW	1'b0
[11:7]	region1_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[6]	region1_pag_en	Enables the CPA mode for non-hashed memory region 1	RW	1'b0
[5:1]	region0_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[0]	region0_pag_en	Enables the CPA mode for non-hashed memory region 0	RW	1'b0

4.3.16.10 cml_port_aggr_mode_ctrl_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures the CCIX port aggregation modes for all non-hashed memory regions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-3) : 16'h11A0 + #{8 * index}
index(4-6) : 16'h2A00 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-600: cml_port_aggr_mode_ctrl_reg1-6

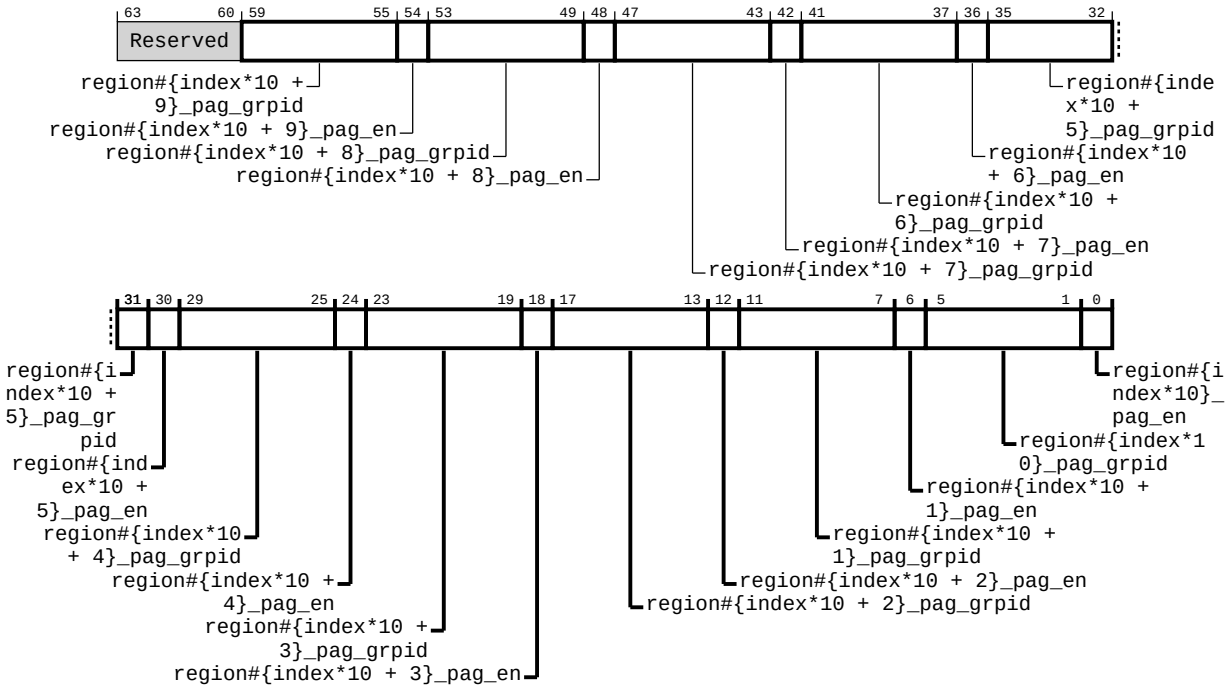


Table 4-616: cml_port_aggr_mode_ctrl_reg1-6 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:55]	region#{index*10 + 9}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[54]	region#{index*10 + 9}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 9}	RW	1'b0
[53:49]	region#{index*10 + 8}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[48]	region#{index*10 + 8}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 8}	RW	1'b0
[47:43]	region#{index*10 + 7}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[42]	region#{index*10 + 7}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 7}	RW	1'b0
[41:37]	region#{index*10 + 6}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[36]	region#{index*10 + 6}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 6}	RW	1'b0
[35:31]	region#{index*10 + 5}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[30]	region#{index*10 + 5}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 5}	RW	1'b0
[29:25]	region#{index*10 + 4}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[24]	region#{index*10 + 4}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 4}	RW	1'b0
[23:19]	region#{index*10 + 3}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[18]	region#{index*10 + 3}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 3}	RW	1'b0
[17:13]	region#{index*10 + 2}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[12]	region#{index*10 + 2}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 2}	RW	1'b0
[11:7]	region#{index*10 + 1}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[6]	region#{index*10 + 1}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10 + 1}	RW	1'b0

Bits	Name	Description	Type	Reset
[5:1]	region#{index*10}_pag_grpid	Specifies CCIX port aggregation group ID	RW	5'h0
[0]	region#{index*10}_pag_en	Enables the CPA mode for non-hashed memory region #{index*10}	RW	1'b0

4.3.16.11 sys_cache_grp_region0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE00 + #{8*index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-601: sys_cache_grp_region0-3

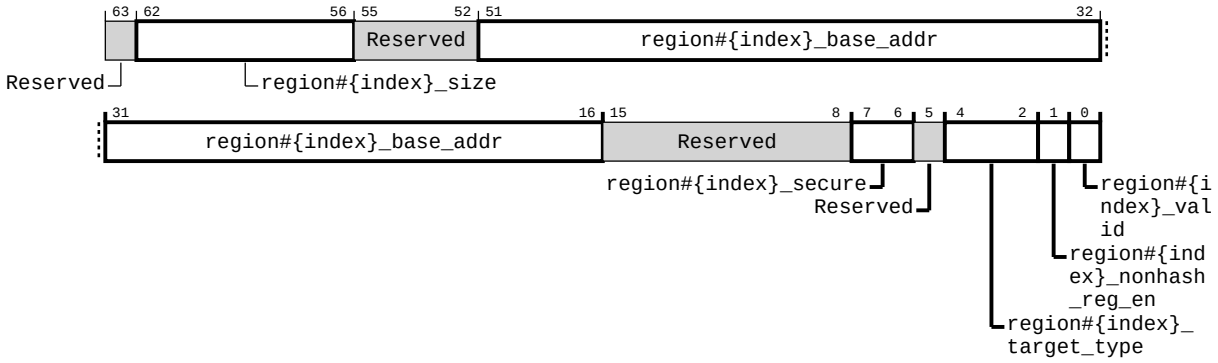


Table 4-617: sys_cache_grp_region0-3 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	region#{index}_size	Memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b0000000
[55:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'b000000000000000000000000000000
[15:8]	Reserved	Reserved	RO	-
[7:6]	region#{index}_secure	Indicates Secure type **2'b00** Trusted device. **2'b01** Trusted device attached memory range only **2'b10** Untrusted device **2'b11** Reserved	RW	2'b10
[5]	Reserved	Reserved	RO	-
[4:2]	region#{index}_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	1'b0
[0]	region#{index}_valid	Memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.12 hashed_tgt_grp_cfg1_region4-31

There are 28 iterations of this register. The index ranges from 4 to 31. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'hE00 + #{8 * index}
index(8-31) : 16'h3000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-602: hashed_tgt_grp_cfg1_region4-31

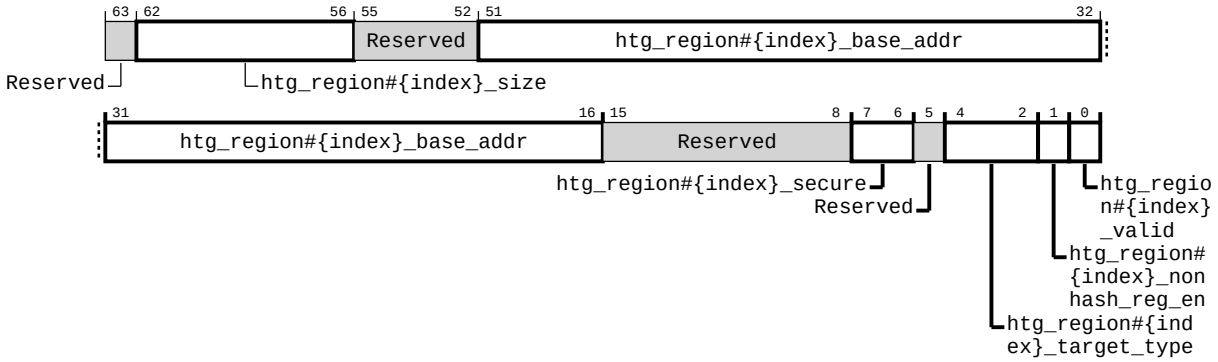


Table 4-618: hashed_tgt_grp_cfg1_region4-31 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_region#{index}_size	Memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b0000000
[55:52]	Reserved	Reserved	RO	-
[51:16]	htg_region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:8]	Reserved	Reserved	RO	-
[7:6]	htg_region#{index}_secure	Indicates Secure type **2'b00** Trusted device. **2'b01** Trusted device attached memory range only **2'b10** Untrusted device **2'b11** Reserved	RW	2'b10
[5]	Reserved	Reserved	RO	-
[4:2]	htg_region#{index}_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	htg_region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	1'b0
[0]	htg_region#{index}_valid	Memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.13 hashed_tgt_grp_cfg2_region0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h3100 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-603: hashed_tgt_grp_cfg2_region0-31

Table 4-619: hashed_tgt_grp_cfg2_region0-31 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:0]	Reserved	Reserved	RO	-

4.3.16.14 sys_cache_grp_secondary_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures secondary hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'hE40 + #{8 * index}
index(8-31) : 16'h3200 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-604: sys_cache_grp_secondary_reg0-3

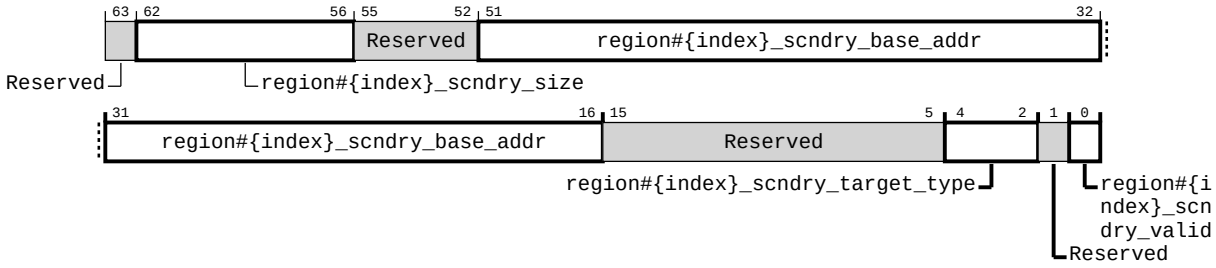


Table 4-620: sys_cache_grp_secondary_reg0-3 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[62:56]	region#{index}_scndry_size	Secondary memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_scndry_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:5]	Reserved	Reserved	RO	-
[4:2]	region#{index}_scndry_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	Reserved	Reserved	RO	-
[0]	region#{index}_scndry_valid	Secondary memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.15 hashed_target_grp_secondary_cfg1_reg4-31

There are 28 iterations of this register. The index ranges from 4 to 31. Configures secondary hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'hE40 + #{8 * index}
index(8-31) : 16'h3200 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-605: hashed_target_grp_secondary_cfg1_reg4-31

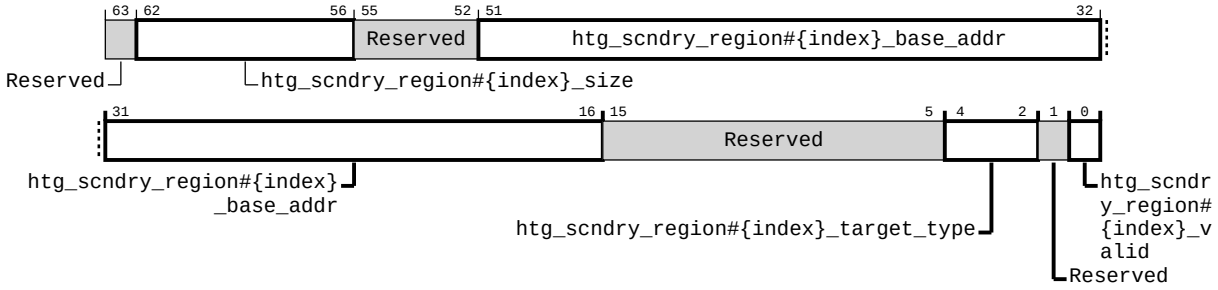


Table 4-621: hashed_target_grp_secondary_cfg1_reg4-31 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_scndry_region#{index}_size	Secondary memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:2]	htg_scndry_region#{index}_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	Reserved	Reserved	RO	-
[0]	htg_scndry_region#{index}_valid	Secondary memory region #{index} valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.16 hashed_target_grp_secondary_cfg2_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h3300 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-606: hashed_target_grp_secondary_cfg2_reg0-31

Table 4-622: hashed_target_grp_secondary_cfg2_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'b000000000000000000000000000000
[15:0]	Reserved	Reserved	RO	-

4.3.16.17 hashed_target_grp_hash_cntl_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures HTG hash type

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h3400 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-607: hashed_target_grp_hash_cntl_reg0-31

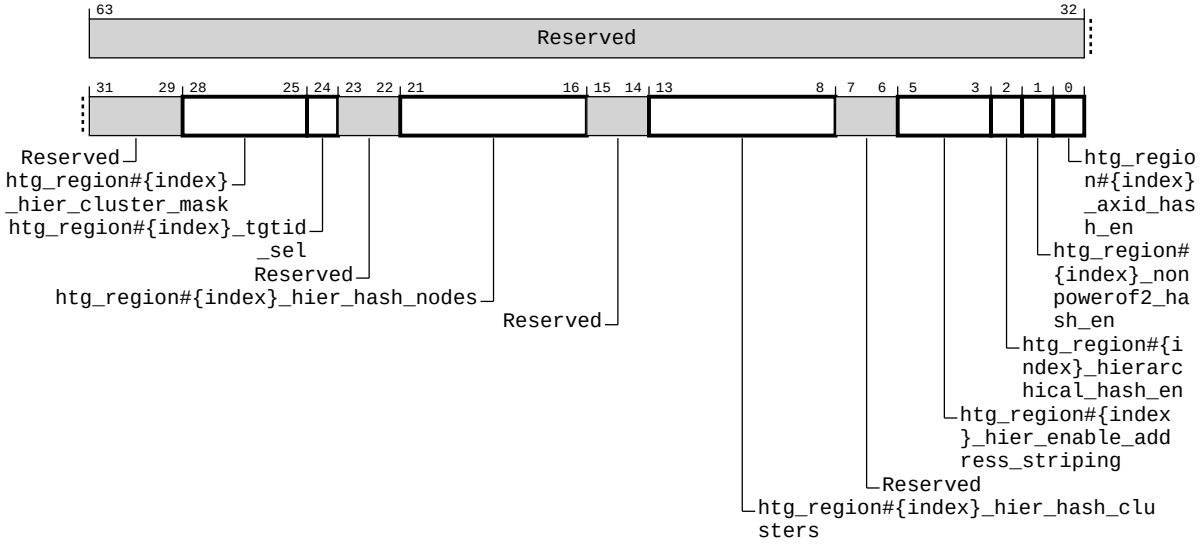


Table 4-623: hashed_target_grp_hash_cntl_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:25]	htg_region#{index}_hier_cluster_mask	Hierarchical hashing: Enable cluster masking to achieve different interleave granularity across clusters. **4'b0000** 64 byte interleave granularity across clusters **4'b0110** 4096 byte interleave granularity across clusters **4'b1111** Cluster interleaving disabled **Others** Reserved	RW	4'b1111
[24]	htg_region#{index}_tgtid_sel	Select the TgtID's from HNF or HNP trgt tables **1'b0** Default, selects from HNF table **1'b1** Selects from HNP table	RW	1'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	htg_region#{index}_hier_hash_nodes	Hierarchical hashing mode, define number of nodes in each cluster	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg_region#{index}_hier_hash_clusters	Hierarchical hashing mode, define number of clusters groups	RW	6'h0

Bits	Name	Description	Type	Reset
[7:6]	Reserved	Reserved	RO	-
[5:3]	htg_region#{index}_hier_enable_address_striping	Hierarchical hashing: configure number of address bits needs to shuttered (removed) at second hierarchy hash (LSB bit is based on cluster mask). **3'b000** No address shuttering **3'b001** One addr bit shuttered (2 clusters) **3'b010** Two addr bit shuttered (4 clusters) **3'b011** Three addr bit shuttered (8 clusters) **3'b100** Four addr bit shuttered (16 clusters) **3'b101** Five addr bit shuttered (32 clusters) **Others** Reserved	RW	3'b0
[2]	htg_region#{index}_hierarchical_hash_en	Hierarchical Hashing mode enable configure bit	RW	1'b0
[1]	htg_region#{index}_nonpowerof2_hash_en	Non power of two Hashing mode enable cconfigure bit	RW	1'b0
[0]	htg_region#{index}_axid_hash_en	AXID based Hashing mode enable configure bit	RW	1'b0

4.3.16.18 sys_cache_group_hn_count

Indicates number of HN-F/HN-P's in hashed target groups 0 to 7.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEA0`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-608: sys_cache_group_hn_count

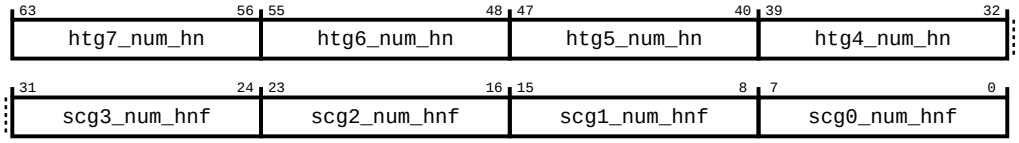


Table 4-624: sys_cache_group_hn_count attributes

Bits	Name	Description	Type	Reset
[63:56]	htg7_num_hn	HN count for hashed target group 7	RW	8'h00
[55:48]	htg6_num_hn	HN count for hashed target group 6	RW	8'h00
[47:40]	htg5_num_hn	HN count for hashed target group 5	RW	8'h00
[39:32]	htg4_num_hn	HN count for hashed target group 4	RW	8'h00
[31:24]	scg3_num_hnf	HN count for hashed target group 3	RW	8'h00
[23:16]	scg2_num_hnf	HN count for hashed target group 2	RW	8'h00
[15:8]	scg1_num_hnf	HN count for hashed target group 1	RW	8'h00
[7:0]	scg0_num_hnf	HN count for hashed target group 0	RW	8'h00

4.3.16.19 hashed_target_group_hn_count_reg1-3

There are 3 iterations of this register. The index ranges from 1 to 3. Indicates number of HN-F/HN-P's in hashed target groups #{index*8} to #{index*8 + 7}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-1) : 16'hEA0 + #{8 * index}
index(2-3) : 16'h3700 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-609: hashed_target_group_hn_count_reg1-3

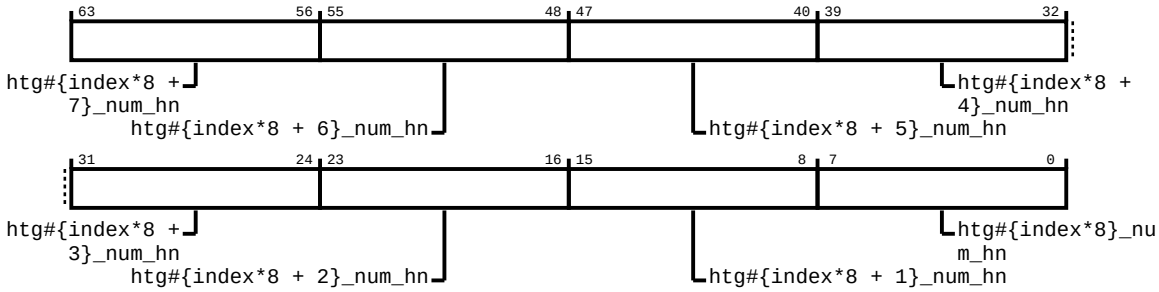


Table 4-625: hashed_target_group_hn_count_reg1-3 attributes

Bits	Name	Description	Type	Reset
[63:56]	htg#{index*8 + 7}_num_hn	HN count for hashed target group 7	RW	8'h00
[55:48]	htg#{index*8 + 6}_num_hn	HN count for hashed target group 6	RW	8'h00
[47:40]	htg#{index*8 + 5}_num_hn	HN count for hashed target group 5	RW	8'h00
[39:32]	htg#{index*8 + 4}_num_hn	HN count for hashed target group 4	RW	8'h00
[31:24]	htg#{index*8 + 3}_num_hn	HN count for hashed target group 3	RW	8'h00
[23:16]	htg#{index*8 + 2}_num_hn	HN count for hashed target group 2	RW	8'h00
[15:8]	htg#{index*8 + 1}_num_hn	HN count for hashed target group 1	RW	8'h00
[7:0]	htg#{index*8}_num_hn	HN count for hashed target group 0	RW	8'h00

4.3.16.20 sys_cache_grp_nonhash_nodeid

Configures non-hashed node IDs for hashed target groups 1 to 5. NOTE: Only applicable in the non-hashed mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEC0`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-610: sys_cache_grp_nonhash_nodeid

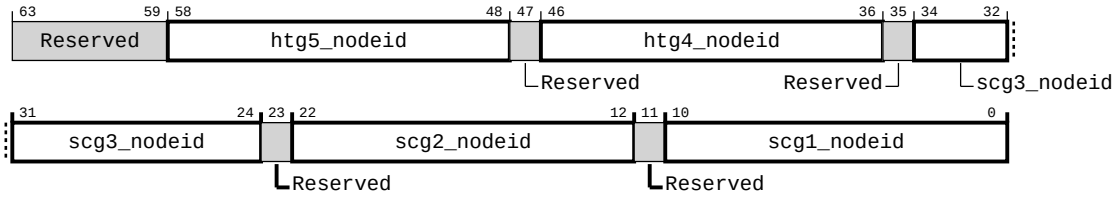


Table 4-626: sys_cache_grp_nonhash_nodeid attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	htg5_nodeid	Non-hashed node ID for Hashed target group 5	RW	11'b00000000000
[47]	Reserved	Reserved	RO	-
[46:36]	htg4_nodeid	Non-hashed node ID for Hashed target group 4	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	scg3_nodeid	Non-hashed node ID for Hashed target group 3	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	scg2_nodeid	Non-hashed node ID for Hashed target group 2	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	scg1_nodeid	Non-hashed node ID for Hashed target group 1	RW	11'b00000000000

4.3.16.21 hashed_target_grp_nonhash_nodeid_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures non-hashed node IDs for hashed target groups #{index*5 + 1} to #{index*5 + 5}. NOTE: Only applicable in the non-hashed mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-5) : 16'hEC0 + #{8 * index}
```

```
index(6-9) : 16'h3800 + #{8 * (index-6)}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-611: hashed_target_grp_nonhash_nodeid_reg1-6

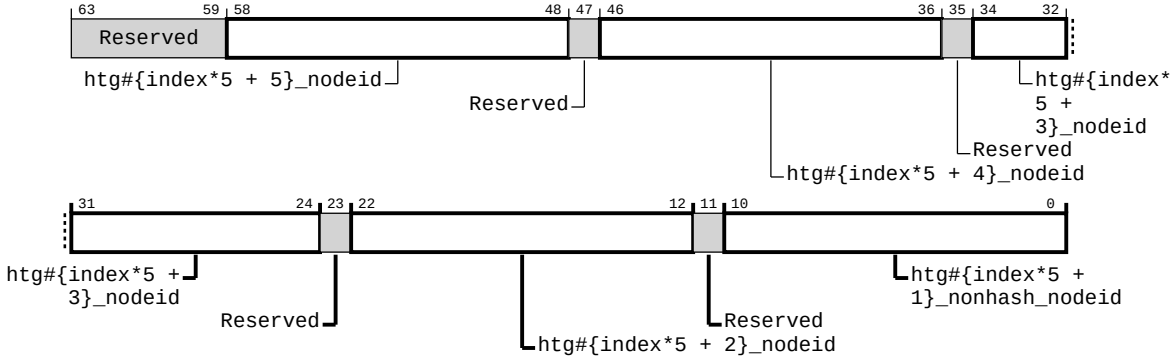


Table 4-627: hashed_target_grp_nonhash_nodeid_reg1-6 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	htg#{index*5 + 5}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 5}	RW	11'b00000000000
[47]	Reserved	Reserved	RO	-
[46:36]	htg#{index*5 + 4}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 4}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	htg#{index*5 + 3}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 3}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	htg#{index*5 + 2}_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 2}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	htg#{index*5 + 1}_nonhash_nodeid	Non-hashed node ID for Hashed target group #{index*5 + 1}	RW	11'b00000000000

4.3.16.22 sys_cache_grp_hn_nodeid_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures HNF node IDs for hashed target groups. Controls target HNF node IDs #{index*4} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`index(0-31) : 16'hF00 + #{8 * index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-612: sys_cache_grp_hn_nodeid_reg0-15

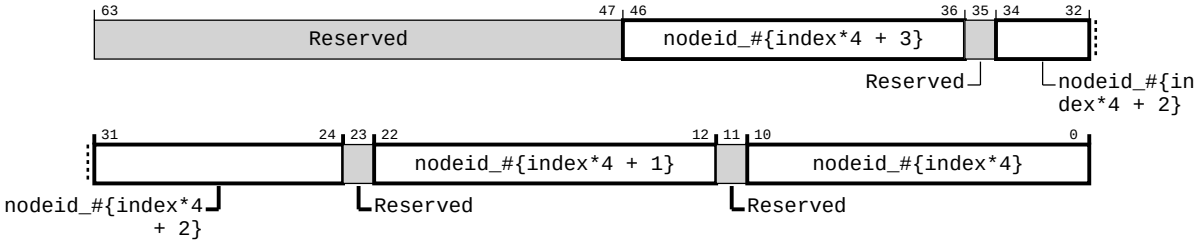


Table 4-628: sys_cache_grp_hn_nodeid_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_#{index*4 + 3}	HNF target node ID #{index*4 + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_#{index*4 + 2}	HNF target node ID #{index*4 + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_#{index*4 + 1}	HNF target node ID #{index*4 + 1}	RW	11'b00000000000

Bits	Name	Description	Type	Reset
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_#{index*4}	HNF target node ID #{index*4}	RW	11'b00000000000

4.3.16.23 hashed_target_grp_hnf_nodeid_reg16-31

There are 16 iterations of this register. The index ranges from 16 to 31. Configures HNF node IDs for hashed target groups. Controls target HNF node IDs #{index*4} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'hF00 + #{8 * index}
index(32-63) : 16'h3500 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-613: hashed_target_grp_hnf_nodeid_reg16-31

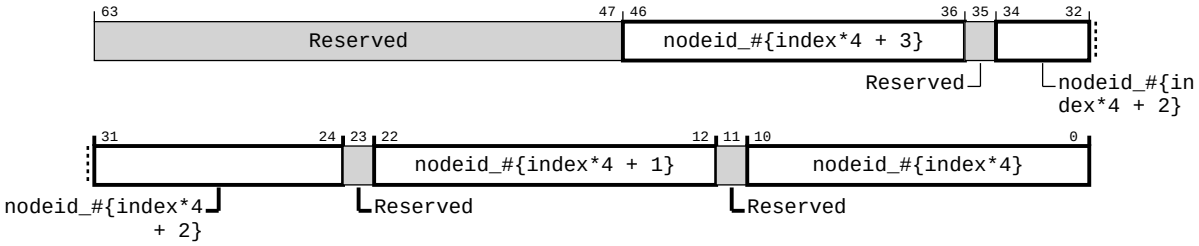


Table 4-629: hashed_target_grp_hnf_nodeid_reg16-31 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[46:36]	nodeid_#{index*4 + 3}	HNF target node ID #{index*4 + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_#{index*4 + 2}	HNF target node ID #{index*4 + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_#{index*4 + 1}	HNF target node ID #{index*4 + 1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_#{index*4}	HNF target node ID #{index*4}	RW	11'b00000000000

4.3.16.24 hashed_target_grp_hnp_nodeid_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures HNP node IDs for hashed target groups. Controls target HNP node IDs #{index*4} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h3600 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-614: hashed_target_grp_hnp_nodeid_reg0-15

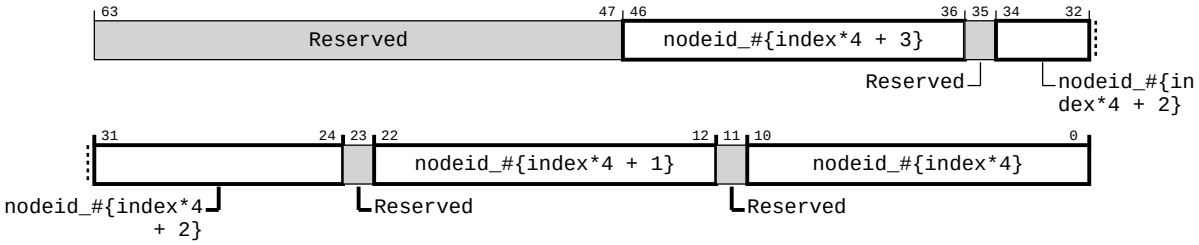


Table 4-630: hashed_target_grp_hnp_nodeid_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_#{index*4 + 3}	HNP target node ID #{index*4 + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_#{index*4 + 2}	HNP target node ID #{index*4 + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_#{index*4 + 1}	HNP target node ID #{index*4 + 1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_#{index*4}	HNP target node ID #{index*4}	RW	11'b00000000000

4.3.16.25 sys_cache_grp_cal_mode_reg

Configures the HN CAL mode support for all hashed target groups.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1120`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-615: sys_cache_grp_cal_mode_reg

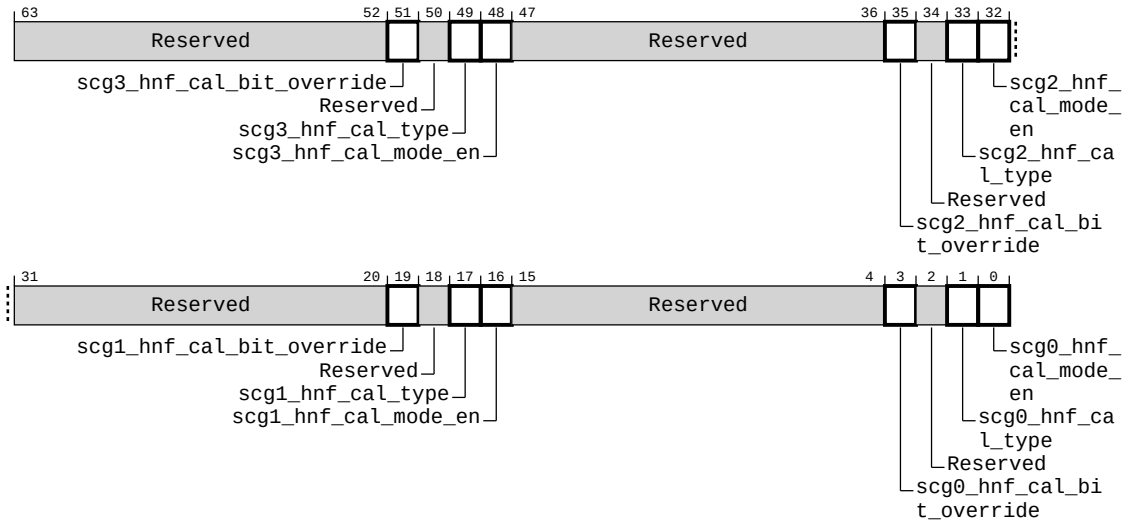


Table 4-631: sys_cache_grp_cal_mode_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	scg3_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 3 **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[50]	Reserved	Reserved	RO	-
[49]	scg3_hnf_cal_type	Enables type of HN CAL for HTG 3 **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[48]	scg3_hnf_cal_mode_en	Enables support for HN CAL for HTG 3	RW	1'b0
[47:36]	Reserved	Reserved	RO	-
[35]	scg2_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 2 **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[34]	Reserved	Reserved	RO	-
[33]	scg2_hnf_cal_type	Enables type of HN CAL for HTG 2 **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[32]	scg2_hnf_cal_mode_en	Enables support for HN CAL for HTG 2	RW	1'b0
[31:20]	Reserved	Reserved	RO	-
[19]	scg1_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 1 **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0

Bits	Name	Description	Type	Reset
[18]	Reserved	Reserved	RO	-
[17]	scg1_hnf_cal_type	Enables type of HN CAL for HTG 1 **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[16]	scg1_hnf_cal_mode_en	Enables support for HN CAL for HTG 1	RW	1'b0
[15:4]	Reserved	Reserved	RO	-
[3]	scg0_hnf_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG 0 **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[2]	Reserved	Reserved	RO	-
[1]	scg0_hnf_cal_type	Enables type of HN CAL for HTG 0 **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[0]	scg0_hnf_cal_mode_en	Enables support for HN CAL for HTG 0	RW	1'b0

4.3.16.26 hashed_target_grp_cal_mode_reg1-7

There are 7 iterations of this register. The index ranges from 1 to 7. Configures the HN CAL mode support for all hashed target groups.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-3) : 16'h1120 + #{8 * index}
index(4-7) : 16'h3780 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-616: hashed_target_grp_cal_mode_reg1-7

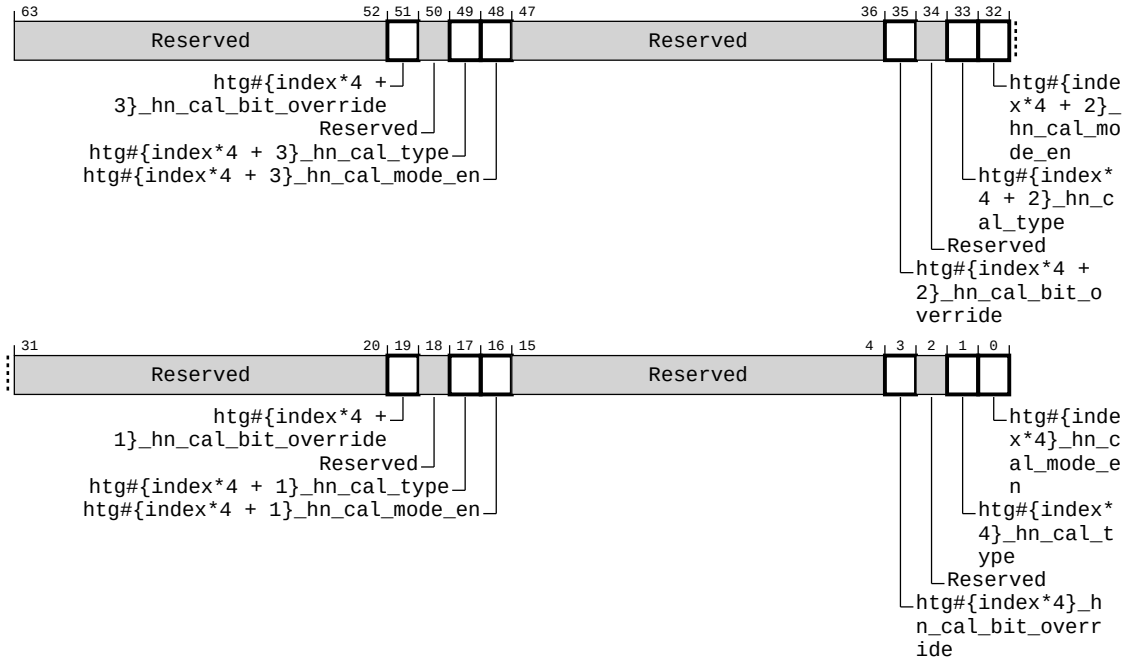


Table 4-632: hashed_target_grp_cal_mode_reg1-7 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	htg#{index*4 + 3}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 3} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[50]	Reserved	Reserved	RO	-
[49]	htg#{index*4 + 3}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 3} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[48]	htg#{index*4 + 3}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 3}	RW	1'b0
[47:36]	Reserved	Reserved	RO	-
[35]	htg#{index*4 + 2}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 2} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[34]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[33]	htg#{index*4 + 2}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 2} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[32]	htg#{index*4 + 2}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 2}	RW	1'b0
[31:20]	Reserved	Reserved	RO	-
[19]	htg#{index*4 + 1}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 1} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[18]	Reserved	Reserved	RO	-
[17]	htg#{index*4 + 1}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 1} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[16]	htg#{index*4 + 1}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 1}	RW	1'b0
[15:4]	Reserved	Reserved	RO	-
[3]	htg#{index*4}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4} **1'b0** Hash MSB bit to override Device ID **1'b1** Hash LSB bit to override Device ID	RW	1'b0
[2]	Reserved	Reserved	RO	-
[1]	htg#{index*4}_hn_cal_type	Enables type of HN CAL for HTG #{index*4} **1'b0** CAL2 mode **1'b1** CAL4 mode	RW	1'b0
[0]	htg#{index*4}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4}	RW	1'b0

4.3.16.27 sys_cache_grp_hn_cpa_en_reg

Configures CCIX port aggregation mode for hashed HNF node IDs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1180`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-617: sys_cache_grp_hn_cpa_en_reg

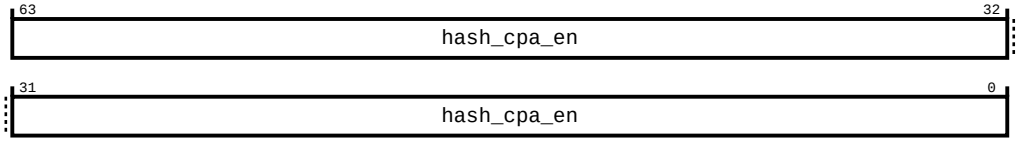


Table 4-633: sys_cache_grp_hn_cpa_en_reg attributes

Bits	Name	Description	Type	Reset
[63:0]	hash_cpa_en	Enable CPA for each hashed HNF node ID	RW	64'h0000000000000000

4.3.16.28 hashed_target_grp_hnf_cpa_en_reg1-1

There are 1 iterations of this register. The index ranges from 1 to 1. Configures CCIX port aggregation mode for hashed HNF node IDs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-1) : 16'h1180 + #{8 * index}
index(2-3) : 16'h3720 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-618: hashed_target_grp_hnf_cpa_en_reg1-1

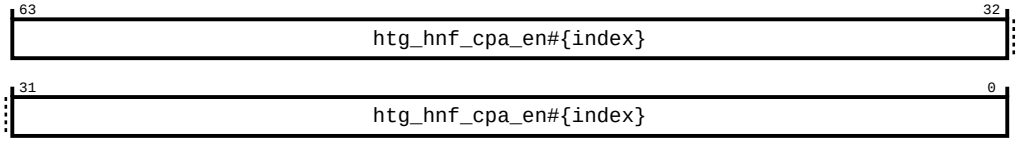


Table 4-634: hashed_target_grp_hnf_cpa_en_reg1-1 attributes

Bits	Name	Description	Type	Reset
[63:0]	htg_hnf_cpa_en#{index}	Enable CPA for each hashed HNF node ID	RW	64'h0000000000000000

4.3.16.29 hashed_target_grp_cpag_perhnf_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures CPAG ID for each hashed HNF node IDs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-15) : 16'h3900 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-619: hashed_target_grp_cpag_perhnf_reg0-15

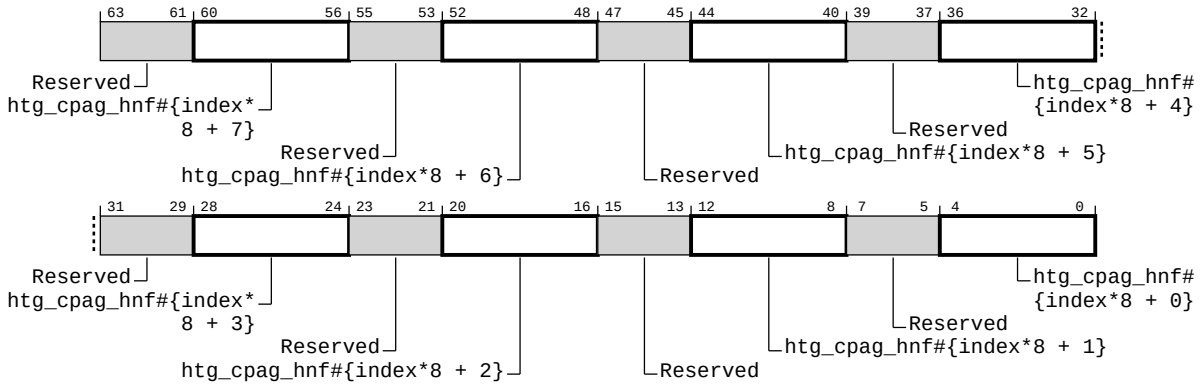


Table 4-635: hashed_target_grp_cpag_perhnf_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:56]	htg_cpag_hnf#{index*8 + 7}	CPAG associated to the HNF#{index*8 + 7}	RW	5'b0
[55:53]	Reserved	Reserved	RO	-
[52:48]	htg_cpag_hnf#{index*8 + 6}	CPAG associated to the HNF#{index*8 + 6}	RW	5'b0
[47:45]	Reserved	Reserved	RO	-
[44:40]	htg_cpag_hnf#{index*8 + 5}	CPAG associated to the HNF#{index*8 + 5}	RW	5'b0
[39:37]	Reserved	Reserved	RO	-
[36:32]	htg_cpag_hnf#{index*8 + 4}	CPAG associated to the HNF#{index*8 + 4}	RW	5'b0
[31:29]	Reserved	Reserved	RO	-
[28:24]	htg_cpag_hnf#{index*8 + 3}	CPAG associated to the HNF#{index*8 + 3}	RW	5'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	htg_cpag_hnf#{index*8 + 2}	CPAG associated to the HNF#{index*8 + 2}	RW	5'b0
[15:13]	Reserved	Reserved	RO	-
[12:8]	htg_cpag_hnf#{index*8 + 1}	CPAG associated to the HNF#{index*8 + 1}	RW	5'b0
[7:5]	Reserved	Reserved	RO	-
[4:0]	htg_cpag_hnf#{index*8 + 0}	CPAG associated to the HNF#{index*8 + 0}	RW	5'b0

4.3.16.30 sys_cache_grp_hn_cpa_grp_reg

Configures CCIX port aggregation group ID for each System Cache Group

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1190

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-620: sys_cache_grp_hn_cpa_grp_reg

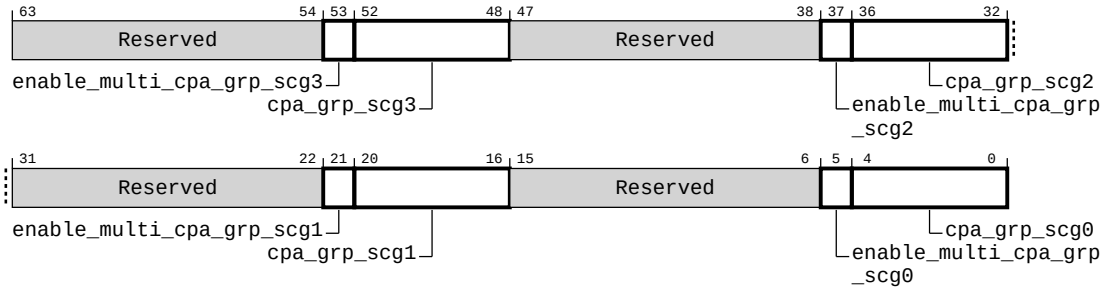


Table 4-636: sys_cache_grp_hn_cpa_grp_reg attributes

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	-
[53]	enable_multi_cpa_grp_scg3	Enables multiple CPA groups to be configured to SCG3 **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to SCG3.	RW	1'b0
[52:48]	cpa_grp_scg3	Specifies CCIX port aggregation group ID for Hashed Target Group 3	RW	5'h0
[47:38]	Reserved	Reserved	RO	-
[37]	enable_multi_cpa_grp_scg2	Enables multiple CPA groups to be configured to SCG2 **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to SCG2.	RW	1'b0
[36:32]	cpa_grp_scg2	Specifies CCIX port aggregation group ID for Hashed target Group 2	RW	5'h0
[31:22]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[21]	enable_multi_cpa_grp_scg1	Enables multiple CPA groups to be configured to SCG1 **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to SCG1.	RW	1'b0
[20:16]	cpa_grp_scg1	Specifies CCIX port aggregation group ID for Hashed target Group 1	RW	5'h0
[15:6]	Reserved	Reserved	RO	-
[5]	enable_multi_cpa_grp_scg0	Enables multiple CPA groups to be configured to SCG0 **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_scg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to SCG0.	RW	1'b0
[4:0]	cpa_grp_scg0	Specifies CCIX port aggregation group ID for hashed target Group 0	RW	5'h0

4.3.16.31 hashed_target_grp_cpa_grp_reg1-7

There are 7 iterations of this register. The index ranges from 1 to 7. Configures CCIX port aggregation group ID for each System Cache Group

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-1) : 16'h1190 + #{8 * index}
index(2-9) : 16'h3740 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-621: hashed_target_grp_cpa_grp_reg1-7

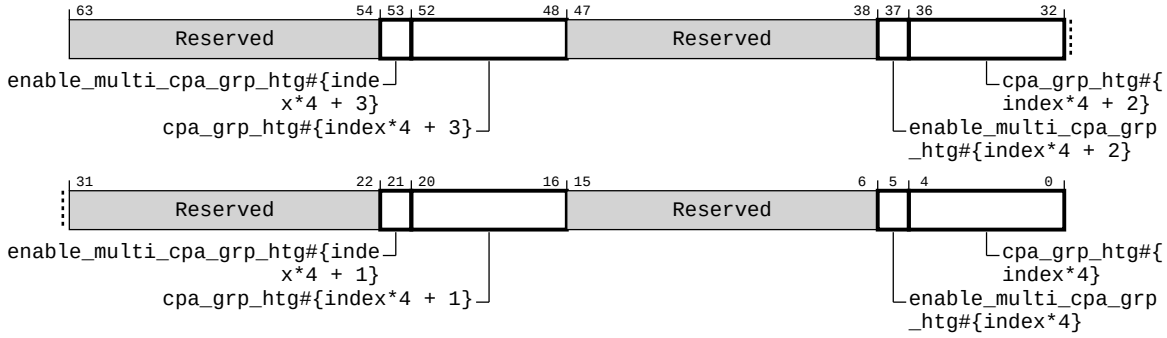


Table 4-637: hashed_target_grp_cpa_grp_reg1-7 attributes

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	-
[53]	enable_multi_cpa_grp_htg#{index*4 + 3}	Enables multiple CPA groups to be configured to HTG **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	1'b0
[52:48]	cpa_grp_htg#{index*4 + 3}	Specifies CCIX port aggregation group ID for Hashed Target Group #{index*4 + 3}	RW	5'h0
[47:38]	Reserved	Reserved	RO	-
[37]	enable_multi_cpa_grp_htg#{index*4 + 2}	Enables multiple CPA groups to be configured to HTG **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	1'b0
[36:32]	cpa_grp_htg#{index*4 + 2}	Specifies CCIX port aggregation group ID for Hashed target Group #{index*4 + 2}	RW	5'h0
[31:22]	Reserved	Reserved	RO	-
[21]	enable_multi_cpa_grp_htg#{index*4 + 1}	Enables multiple CPA groups to be configured to HTG **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	1'b0
[20:16]	cpa_grp_htg#{index*4 + 1}	Specifies CCIX port aggregation group ID for Hashed target Group #{index*4 + 1}	RW	5'h0
[15:6]	Reserved	Reserved	RO	-
[5]	enable_multi_cpa_grp_htg#{index*4}	Enables multiple CPA groups to be configured to HTG **1'b0** Default/Legacy mode: single CPA group ID is configured in cpa_grp_htg **1'b1** multi_cpag mode: multiple CPA group ID's are configured to HTG.	RW	1'b0

Bits	Name	Description	Type	Reset
[4:0]	cpa_grp_htg#{index*4}	Specifies CCIX port aggregation group ID for hashed target Group #{index*4}	RW	5'h0

4.3.16.32 hashed_target_grp_hnf_lcn_bound_cfg_reg0-1

There are 2 iterations of this register. The index ranges from 0 to 1. Configures cache lines routed to the HNF as LCN bound or Home bound 1'b0: cache lines routed to Home bound 1'b1: cache lines routed to LCN bound

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-3) : 16'h37C0 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-622: hashed_target_grp_hnf_lcn_bound_cfg_reg0-1

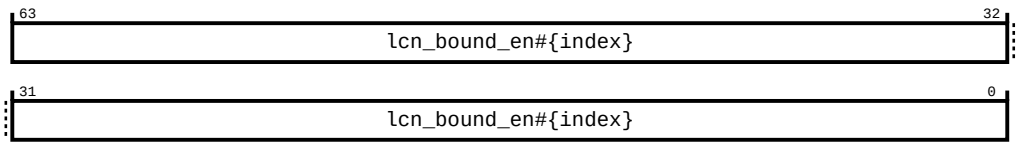


Table 4-638: hashed_target_grp_hnf_lcn_bound_cfg_reg0-1 attributes

Bits	Name	Description	Type	Reset
[63:0]	lcn_bound_en#{index}	Marks the Hashed HNF index as a LCN	RW	64'h0000000000000000

4.3.16.33 hashed_target_grp_hnf_target_type_override_cfg_reg0-1

There are 2 iterations of this register. The index ranges from 0 to 1. Configures the target type for each targetID, RNI/D uses this information to enable tunneling vs streaming. When POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1, this indication is derived from cpa_en 1'b0: HNF target 1'b1: CCG target

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-1) : 16'h37E0 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-623: hashed_target_grp_hnf_target_type_override_cfg_reg0-1

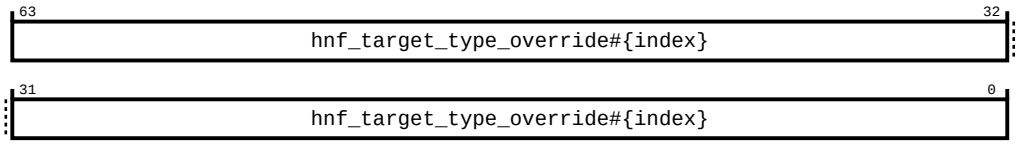


Table 4-639: hashed_target_grp_hnf_target_type_override_cfg_reg0-1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_target_type_override#{index}	Overrides the HNF target type with the CCG	RW	64'h0000000000000000

4.3.16.34 hashed_target_grp_compact_cpag_ctrl0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the CPAG control for HTG#{index} valid only when POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h3A00 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-624: hashed_target_grp_compact_cpag_ctrl0-31

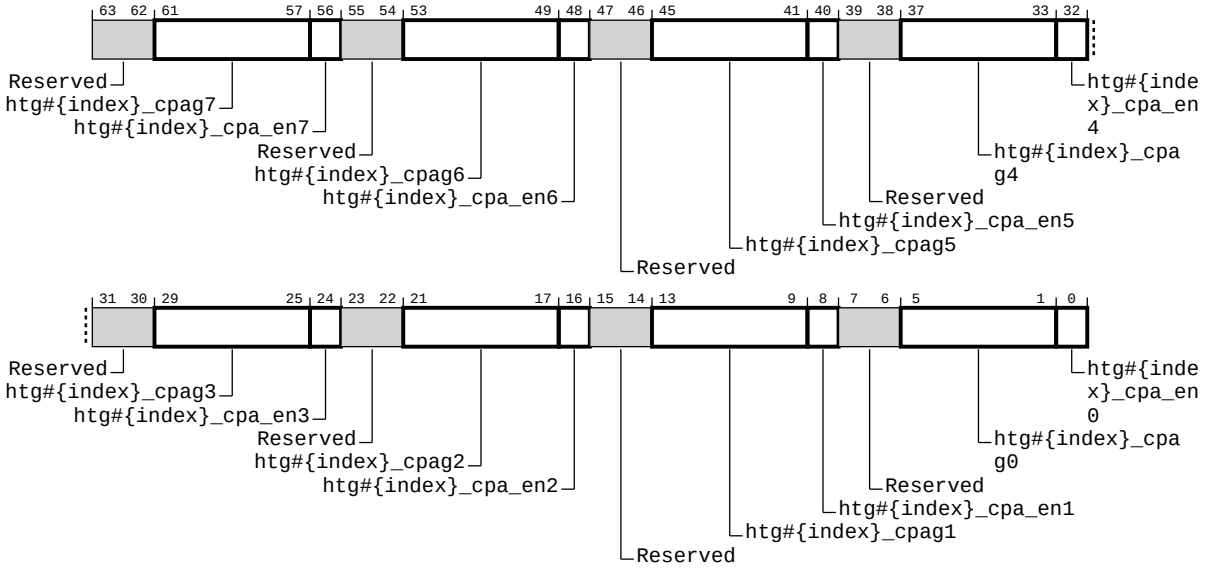


Table 4-640: hashed_target_grp_compact_cpag_ctrl0-31 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:57]	htg#{index}_cpag7	cpag id for index7	RW	5'b0
[56]	htg#{index}_cpa_en7	cpa enable for index7	RW	1'b0
[55:54]	Reserved	Reserved	RO	-
[53:49]	htg#{index}_cpag6	cpag id for index6	RW	5'b0
[48]	htg#{index}_cpa_en6	cpa enable for index6	RW	1'b0
[47:46]	Reserved	Reserved	RO	-
[45:41]	htg#{index}_cpag5	cpag id for index5	RW	5'b0
[40]	htg#{index}_cpa_en5	cpa enable for index5	RW	1'b0
[39:38]	Reserved	Reserved	RO	-
[37:33]	htg#{index}_cpag4	cpag id for index4	RW	5'b0
[32]	htg#{index}_cpa_en4	cpa enable for index4	RW	1'b0
[31:30]	Reserved	Reserved	RO	-
[29:25]	htg#{index}_cpag3	cpag id for index0	RW	5'b0
[24]	htg#{index}_cpa_en3	cpa enable for index3	RW	1'b0
[23:22]	Reserved	Reserved	RO	-
[21:17]	htg#{index}_cpag2	cpag id for index2	RW	5'b0
[16]	htg#{index}_cpa_en2	cpa enable for index2	RW	1'b0
[15:14]	Reserved	Reserved	RO	-
[13:9]	htg#{index}_cpag1	cpag id for index1	RW	5'b0
[8]	htg#{index}_cpa_en1	cpa enable for index1	RW	1'b0
[7:6]	Reserved	Reserved	RO	-
[5:1]	htg#{index}_cpag0	cpag id for index0	RW	5'b0
[0]	htg#{index}_cpa_en0	cpa enable for index0	RW	1'b0

4.3.16.35 hashed_target_grp_compact_hash_ctrl0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the HNF hash selecetion and CPAG hash selection control information for HTG#{index} valid only when POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h3B00 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-625: hashed_target_grp_compact_hash_ctrl0-31

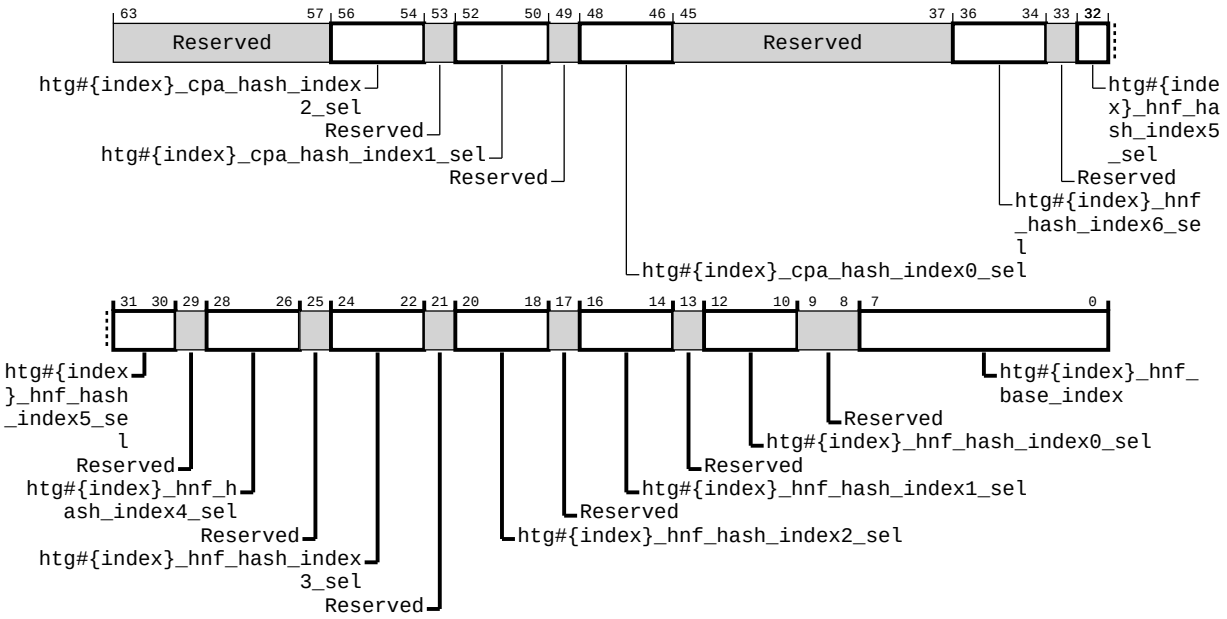


Table 4-641: hashed_target_grp_compact_hash_ctrl0-31 attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:54]	htg#{index}_cpa_hash_index2_sel	configures the CPAG hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index2. **3'b001** SMP hash index2 + 1. **3'b010** SMP hash index2 + 2. **3'b011** SMP hash index2 + 3. **3'b100** SMP hash index2 + 4. **3'b101** SMP hash index2 + 5 **3'b110** SMP hash index2 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[53]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[52:50]	htg#{index}_cpa_hash_index1_sel	configures the CPAG hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index1. **3'b001** SMP hash index1 + 1. **3'b010** SMP hash index1 + 2. **3'b011** SMP hash index1 + 3. **3'b100** SMP hash index1 + 4. **3'b101** SMP hash index1 + 5 **3'b110** SMP hash index1 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[49]	Reserved	Reserved	RO	-
[48:46]	htg#{index}_cpa_hash_index0_sel	configures the CPAG hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index0. **3'b001** SMP hash index0 + 1. **3'b010** SMP hash index0 + 2. **3'b011** SMP hash index0 + 3. **3'b100** SMP hash index0 + 4. **3'b101** SMP hash index0 + 5 **3'b110** SMP hash index0 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[45:37]	Reserved	Reserved	RO	-
[36:34]	htg#{index}_hnf_hash_index6_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index6. **3'b001** SMP hash index6 + 1. **3'b010** SMP hash index6 + 2. **3'b011** SMP hash index6 + 3. **3'b100** SMP hash index6 + 4. **3'b101** SMP hash index6 + 5 **3'b110** SMP hash index6 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[33]	Reserved	Reserved	RO	-
[32:30]	htg#{index}_hnf_hash_index5_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index5. **3'b001** SMP hash index5 + 1. **3'b010** SMP hash index5 + 2. **3'b011** SMP hash index5 + 3. **3'b100** SMP hash index5 + 4. **3'b101** SMP hash index5 + 5 **3'b110** SMP hash index5 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[29]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[28:26]	htg#{index}_hnf_hash_index4_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index4. **3'b001** SMP hash index4 + 1. **3'b010** SMP hash index4 + 2. **3'b011** SMP hash index4 + 3. **3'b100** SMP hash index4 + 4. **3'b101** SMP hash index4 + 5 **3'b110** SMP hash index4 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[25]	Reserved	Reserved	RO	-
[24:22]	htg#{index}_hnf_hash_index3_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index3. **3'b001** SMP hash index3 + 1. **3'b010** SMP hash index3 + 2. **3'b011** SMP hash index3 + 3. **3'b100** SMP hash index3 + 4. **3'b101** SMP hash index3 + 5 **3'b110** SMP hash index3 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[21]	Reserved	Reserved	RO	-
[20:18]	htg#{index}_hnf_hash_index2_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index2. **3'b001** SMP hash index2 + 1. **3'b010** SMP hash index2 + 2. **3'b011** SMP hash index2 + 3. **3'b100** SMP hash index2 + 4. **3'b101** SMP hash index2 + 5 **3'b110** SMP hash index2 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[17]	Reserved	Reserved	RO	-
[16:14]	htg#{index}_hnf_hash_index1_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index1. **3'b001** SMP hash index1 + 1. **3'b010** SMP hash index1 + 2. **3'b011** SMP hash index1 + 3. **3'b100** SMP hash index1 + 4. **3'b101** SMP hash index1 + 5 **3'b110** SMP hash index1 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[13]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[12:10]	htg#{index}_hnf_hash_index0_sel	configures the local hnfs hash selection bits from the total hnfs hash across SMP. **3'b000** pass through from the SMP hnf_hash_index0. **3'b001** SMP hash index0 + 1. **3'b010** SMP hash index0 + 2. **3'b011** SMP hash index0 + 3. **3'b100** SMP hash index0 + 4. **3'b101** SMP hash index0 + 5 **3'b110** SMP hash index0 + 6 **3'b111** Hardcoded value (1'b0)	RW	3'b0
[9:8]	Reserved	Reserved	RO	-
[7:0]	htg#{index}_hnf_base_index	base index to the HNF target ID table	RW	8'b0

4.3.16.36 rnsam_hash_addr_mask_reg

Configures the address mask that is applied before hashing the address bits.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE80`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-626: rnsam_hash_addr_mask_reg

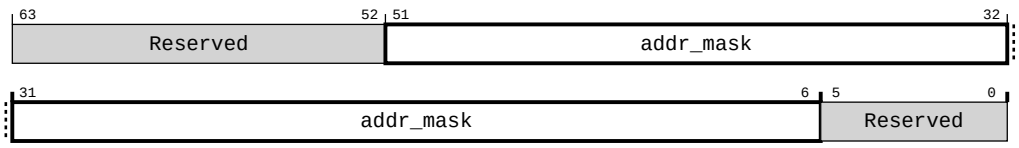


Table 4-642: rnsam_hash_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address mask applied before hashing	RW	46'h3FFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

4.3.16.37 rnsam_hash_axi_id_mask_reg

Configures the AXI_ID mask that is applied before hashing the address bits.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE88`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-627: rnsam_hash_axi_id_mask_reg

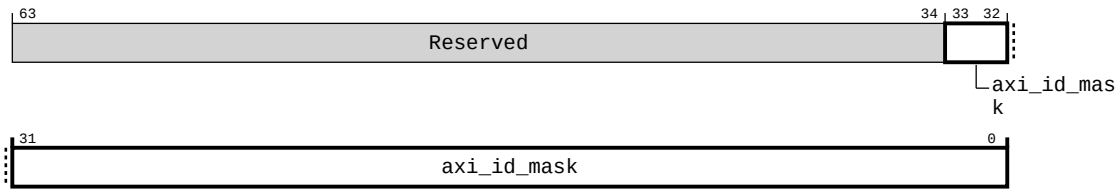


Table 4-643: rnsam_hash_axi_id_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:34]	Reserved	Reserved	RO	-
[33:0]	axi_id_mask	AXI_ID mask applied before hashing	RW	34'h3FFFFFFFF

4.3.16.38 rnsam_region_cmp_addr_mask_reg

Configures the address mask that is applied before region compare.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hE90`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-628: rnsam_region_cmp_addr_mask_reg

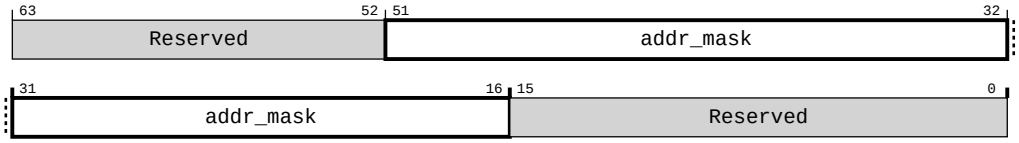


Table 4-644: rnsam_region_cmp_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	addr_mask	Address mask applied before memory region compare	RW	36'hFFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

4.3.16.39 cml_port_aggr_grp0-31_add_mask

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the CCIX port aggregation address mask for group #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-5) : 16'h11C0 + #{8 * index}
index(6-37) : 16'h2B00 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-629: cml_port_aggr_grp0-31_add_mask

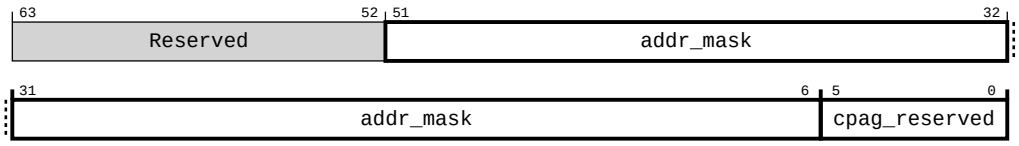


Table 4-645: cml_port_aggr_grp0-31_add_mask attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address/AXID mask to be applied before hashing CONSTRAINT: ADDR MASK is [51:6] CONSTRAINT: AXID MASK is [31:0] = [37:6]	RW	46'h3FFFFFFFFFFF
[5:0]	cpag_reserved	reserved bits	RW	6'h0

4.3.16.40 cml_cpag_base_indx_grp0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures the CPAG base indexes.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'h2B00 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-630: cml_cpag_base_indx_grp0-3

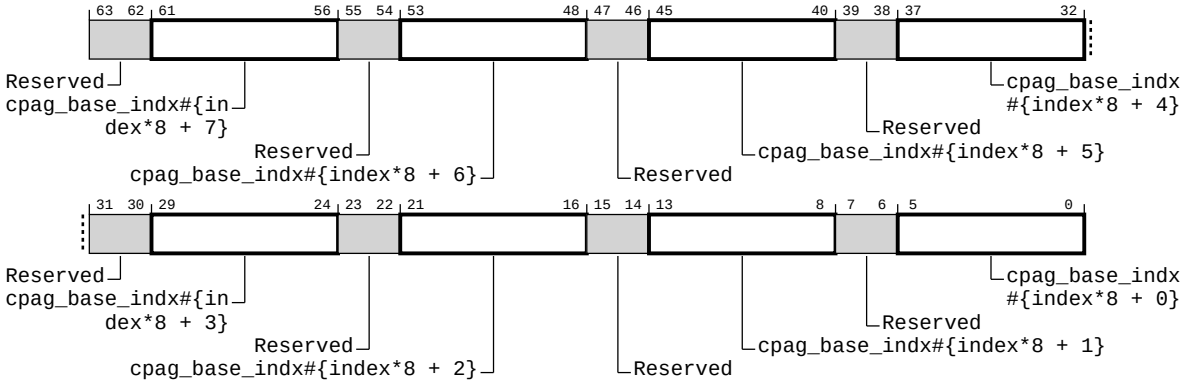


Table 4-646: cml_cpag_base_indx_grp0-3 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	cpag_base_indx#{index*8 + 7}	Configures the CPAG base index #{8*index + 7}	RW	6'h3F
[55:54]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[53:48]	cpag_base_indx#{index*8 + 6}	Configures the CPAG base index #{8*index + 6}	RW	6'h3F
[47:46]	Reserved	Reserved	RO	-
[45:40]	cpag_base_indx#{index*8 + 5}	Configures the CPAG base index #{8*index + 5}	RW	6'h3F
[39:38]	Reserved	Reserved	RO	-
[37:32]	cpag_base_indx#{index*8 + 4}	Configures the CPAG base index #{8*index + 4}	RW	6'h3F
[31:30]	Reserved	Reserved	RO	-
[29:24]	cpag_base_indx#{index*8 + 3}	Configures the CPAG base index #{8*index + 3}	RW	6'h3F
[23:22]	Reserved	Reserved	RO	-
[21:16]	cpag_base_indx#{index*8 + 2}	Configures the CPAG base index #{8*index + 2}	RW	6'h3F
[15:14]	Reserved	Reserved	RO	-
[13:8]	cpag_base_indx#{index*8 + 1}	Configures the CPAG base index #{8*index + 1}	RW	6'h3F
[7:6]	Reserved	Reserved	RO	-
[5:0]	cpag_base_indx#{index*8 + 0}	Configures the CPAG base index #{8*index + 0}	RW	6'h3F

4.3.16.41 cml_port_aggr_grp_reg0-12

There are 13 iterations of this register. The index ranges from 0 to 12. Configures the CCIX port aggregation port Node IDs

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-2) : 16'h11F0 + #{8 * index}
index(3-14) : 16'h2C00 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-631: cml_port_aggr_grp_reg0-12

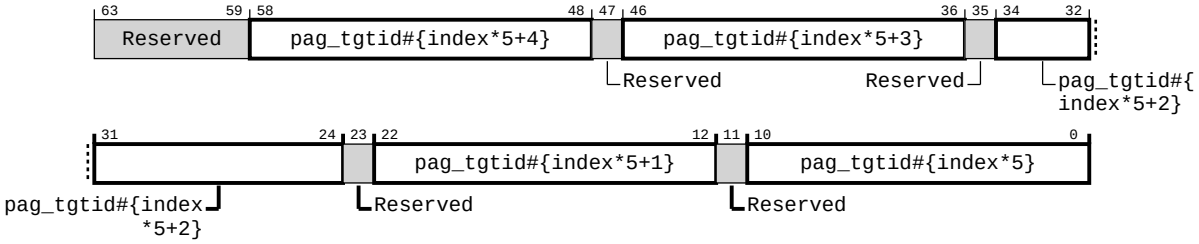


Table 4-647: cml_port_aggr_grp_reg0-12 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	pag_tgtid#{index*5+4}	Specifies target ID #{index*5+4} for CPAG	RW	11'b0
[47]	Reserved	Reserved	RO	-
[46:36]	pag_tgtid#{index*5+3}	Specifies target ID #{index*5+3} for CPAG	RW	11'b0
[35]	Reserved	Reserved	RO	-
[34:24]	pag_tgtid#{index*5+2}	Specifies target ID #{index*5+2} for CPAG	RW	11'b0
[23]	Reserved	Reserved	RO	-
[22:12]	pag_tgtid#{index*5+1}	Specifies target ID #{index*5+1} for CPAG	RW	11'b0
[11]	Reserved	Reserved	RO	-
[10:0]	pag_tgtid#{index*5}	Specifies target ID #{index*5} for CPAG	RW	11'b0

4.3.16.42 cml_port_aggr_ctrl_reg

Configures the CCIX port aggregation port IDs for group 2.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1208`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-632: cml_port_aggr_ctrl_reg

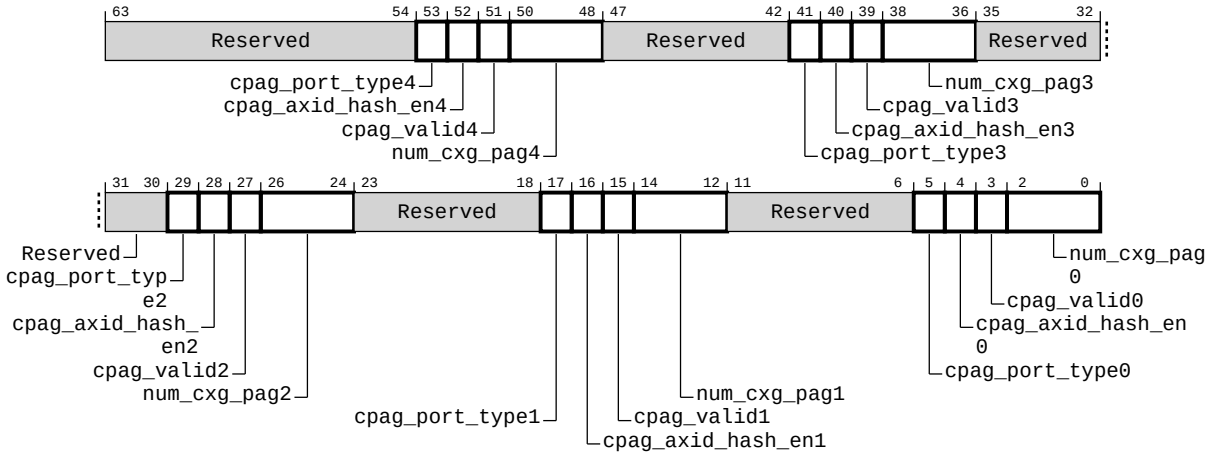


Table 4-648: cml_port_aggr_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	-
[53]	cpag_port_type4	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[52]	cpag_axid_hash_en4	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[51]	cpag_valid4	Valid programming for CPAG4, Enabled by default (backward compatible)	RW	1'b1
[50:48]	num_cxg_pag4	Specifies the number of CXRAs in CPAG4 **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[47:42]	Reserved	Reserved	RO	-
[41]	cpag_port_type3	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0

Bits	Name	Description	Type	Reset
[40]	cpag_axid_hash_en3	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[39]	cpag_valid3	Valid programming for CPAG + 3}, Enabled by default (backward compatible)	RW	1'b1
[38:36]	num_cxg_pag3	Specifies the number of CXRAs in CPAG3 **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[35:30]	Reserved	Reserved	RO	-
[29]	cpag_port_type2	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[28]	cpag_axid_hash_en2	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[27]	cpag_valid2	Valid programming for CPAG + 2}, Enabled by default (backward compatible)	RW	1'b1
[26:24]	num_cxg_pag2	Specifies the number of CXRAs in CPAG2 **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[23:18]	Reserved	Reserved	RO	-
[17]	cpag_port_type1	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[16]	cpag_axid_hash_en1	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[15]	cpag_valid1	Valid programming for CPAG + 1}, Enabled by default (backward compatible)	RW	1'b1
[14:12]	num_cxg_pag1	Specifies the number of CXRAs in CPAG1 **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[11:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5]	cpag_port_type0	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[4]	cpag_axid_hash_en0	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[3]	cpag_valid0	Valid programming for CPAG, Enabled by default (backward compatible)	RW	1'b1
[2:0]	num_cxg_pag0	Specifies the number of CXRAs in CPAG0 **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0

4.3.16.43 cml_port_aggr_ctrl_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures the CCIX port aggregation port IDs for group 2.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-15) : 16'h1208 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-633: cml_port_aggr_ctrl_reg1-6

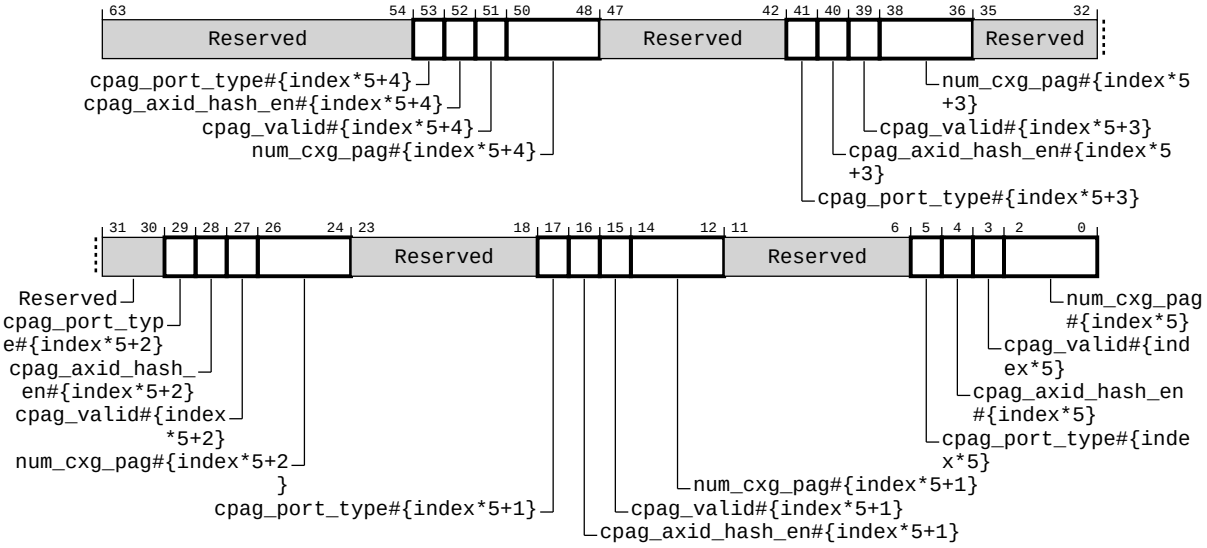


Table 4-649: cml_port_aggr_ctrl_reg1-6 attributes

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	-
[53]	cpag_port_type#{index*5+4}	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[52]	cpag_axid_hash_en#{index*5+4}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[51]	cpag_valid#{index*5+4}	Valid programming for CPAG #{index*5 + 4}, Enabled by default (backward compatible)	RW	1'b1
[50:48]	num_cxg_pag#{index*5+4}	Specifies the number of CXRAs in CPAG #{index*5 + 4} **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[47:42]	Reserved	Reserved	RO	-
[41]	cpag_port_type#{index*5+3}	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[40]	cpag_axid_hash_en#{index*5+3}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[39]	cpag_valid#{index*5+3}	Valid programming for CPAG #{index*5 + 3}, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[38:36]	num_cxg_pag#{index*5+3}	Specifies the number of CXRAs in CPAG #{index*5 + 3} **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[35:30]	Reserved	Reserved	RO	-
[29]	cpag_port_type#{index*5+2}	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[28]	cpag_axid_hash_en#{index*5+2}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[27]	cpag_valid#{index*5+2}	Valid programming for CPAG #{index*5 + 2}, Enabled by default (backward compatible)	RW	1'b1
[26:24]	num_cxg_pag#{index*5+2}	Specifies the number of CXRAs in CPAG #{index*5 + 2} **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[23:18]	Reserved	Reserved	RO	-
[17]	cpag_port_type#{index*5+1}	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0
[16]	cpag_axid_hash_en#{index*5+1}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[15]	cpag_valid#{index*5+1}	Valid programming for CPAG #{index*5 + 1}, Enabled by default (backward compatible)	RW	1'b1
[14:12]	num_cxg_pag#{index*5+1}	Specifies the number of CXRAs in CPAG #{index*5 + 1} **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0
[11:6]	Reserved	Reserved	RO	-
[5]	cpag_port_type#{index*5}	Specifies the port type **1'b0** CXL port **1'b1** CML SMP port	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	cpag_axid_hash_en#{index*5}	Enable AXID based port aggregation, by default address based hashing is enabled	RW	1'b0
[3]	cpag_valid#{index*5}	Valid programming for CPAG #{index*5}, Enabled by default (backward compatible)	RW	1'b1
[2:0]	num_cxg_pag#{index*5}	Specifies the number of CXRAs in CPAG #{index*5} **3'b000** 1 port **3'b001** 2 ports **3'b010** 4 ports **3'b011** 8 ports **3'b100** 16 ports **3'b101** 32 ports **3'b110** 3 ports (MOD-3 hash) **3'b111** Reserved	RW	3'b0

4.3.16.44 sys_cache_grp_sn_attr

Configures attributes for SN node IDs for system cache groups.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEB0`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-634: sys_cache_grp_sn_attr

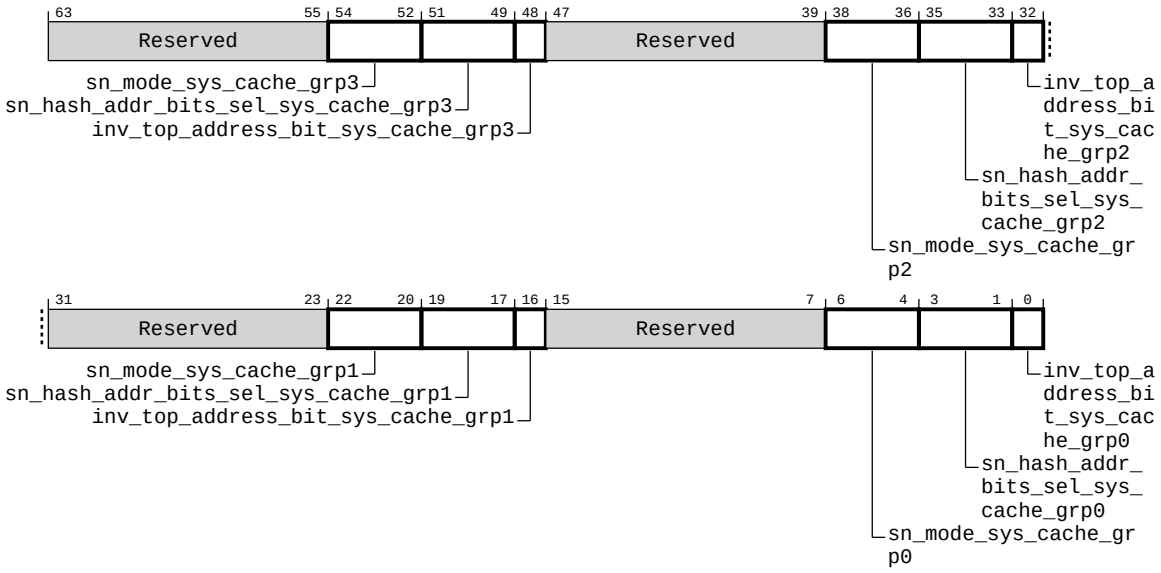


Table 4-650: sys_cache_grp_sn_attr attributes

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:52]	sn_mode_sys_cache_grp3	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b0
[51:49]	sn_hash_addr_bits_sel_sys_cache_grp3	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[48]	inv_top_address_bit_sys_cache_grp3	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[47:39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:36]	sn_mode_sys_cache_grp2	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b00
[35:33]	sn_hash_addr_bits_sel_sys_cache_grp2	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[32]	inv_top_address_bit_sys_cache_grp2	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[31:23]	Reserved	Reserved	RO	-
[22:20]	sn_mode_sys_cache_grp1	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b0
[19:17]	sn_hash_addr_bits_sel_sys_cache_grp1	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[16]	inv_top_address_bit_sys_cache_grp1	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[15:7]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[6:4]	sn_mode_sys_cache_grp0	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b0
[3:1]	sn_hash_addr_bits_sel_sys_cache_grp0	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[0]	inv_top_address_bit_sys_cache_grp0	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0

4.3.16.45 sys_cache_grp_sn_attr1

Configures attributes for SN node IDs for system cache groups.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hEB8`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-635: sys_cache_grp_sn_attr1

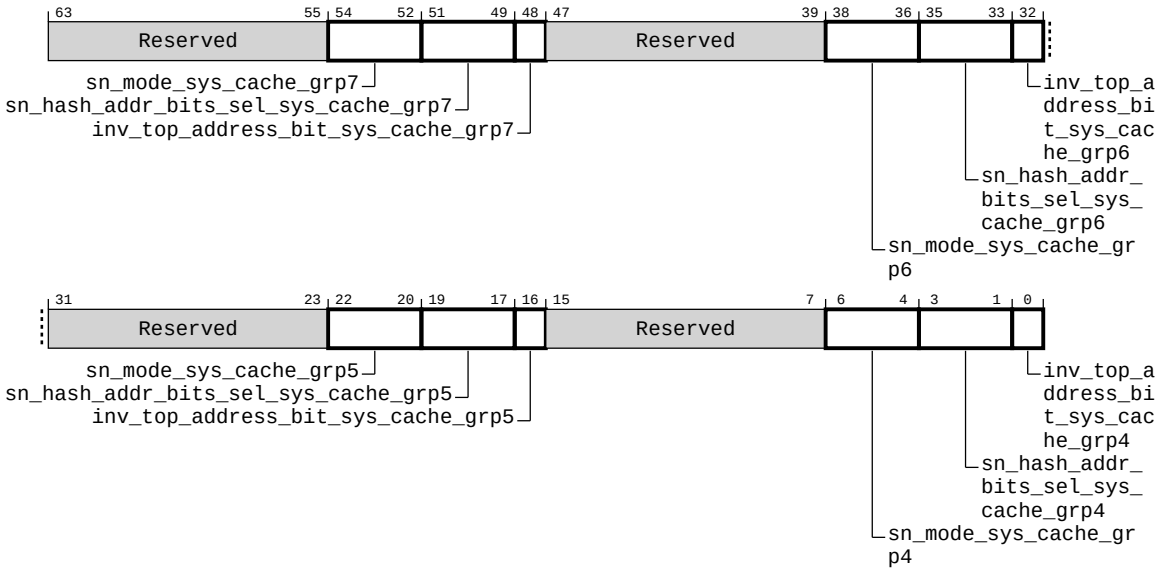


Table 4-651: sys_cache_grp_sn_attr1 attributes

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:52]	sn_mode_sys_cache_grp7	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b0
[51:49]	sn_hash_addr_bits_sel_sys_cache_grp7	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[48]	inv_top_address_bit_sys_cache_grp7	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[47:39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38:36]	sn_mode_sys_cache_grp6	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b00
[35:33]	sn_hash_addr_bits_sel_sys_cache_grp6	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[32]	inv_top_address_bit_sys_cache_grp6	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[31:23]	Reserved	Reserved	RO	-
[22:20]	sn_mode_sys_cache_grp5	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b0
[19:17]	sn_hash_addr_bits_sel_sys_cache_grp5	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[16]	inv_top_address_bit_sys_cache_grp5	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[15:7]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[6:4]	sn_mode_sys_cache_grp4	SN selection mode **3'b000** 1-SN mode (SN0) **3'b001** 3-SN mode (SN0, SN1, SN2) **3'b010** 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) **3'b011** 5-SN mode (SN0, SN1, SN2, SN3, SN4) **3'b100** 2-SN mode (SN0, SN1) power of 2 hashing **3'b101** 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing **3'b110** 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing **Others** Reserved	RW	3'b0
[3:1]	sn_hash_addr_bits_sel_sys_cache_grp4	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[0]	inv_top_address_bit_sys_cache_grp4	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0

4.3.16.46 sys_cache_grp_sn_sam_cfg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures top address bits for SN SAM system cache groups #{index*2} and #{index*2 + 1}. All top_address_bit fields must be between bits 47 and 28. top_address_bit2 > top_address_bit1 > top_address_bit0.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'h1140 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-636: sys_cache_grp_sn_sam_cfg0-3

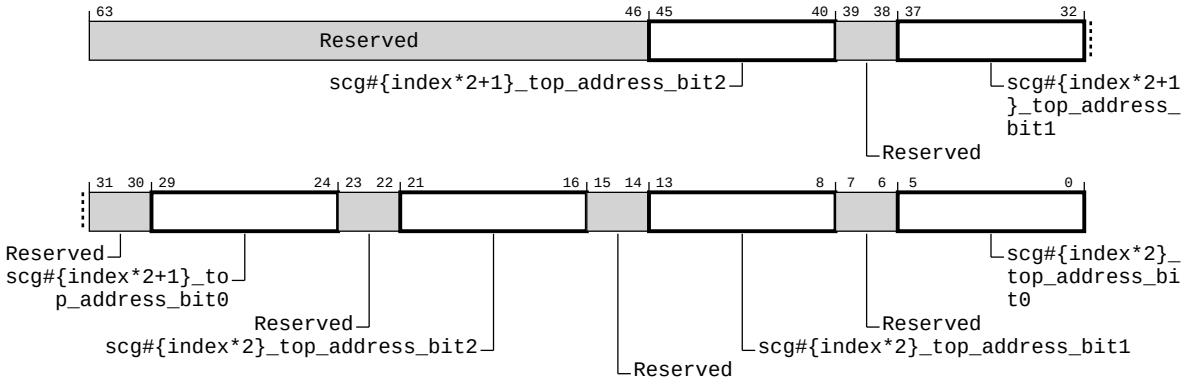


Table 4-652: sys_cache_grp_sn_sam_cfg0-3 attributes

Bits	Name	Description	Type	Reset
[63:46]	Reserved	Reserved	RO	-
[45:40]	scg#{index*2+1}_top_address_bit2	Top address bit 2 for system cache group #{index*2+1}	RW	6'h00
[39:38]	Reserved	Reserved	RO	-
[37:32]	scg#{index*2+1}_top_address_bit1	Top address bit 1 for system cache group #{index*2+1}	RW	6'h00
[31:30]	Reserved	Reserved	RO	-
[29:24]	scg#{index*2+1}_top_address_bit0	Top address bit 0 for system cache group #{index*2+1}	RW	6'h00
[23:22]	Reserved	Reserved	RO	-
[21:16]	scg#{index*2}_top_address_bit2	Top address bit 2 for system cache group #{index*2}	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	scg#{index*2}_top_address_bit1	Top address bit 1 for system cache group #{index*2}	RW	6'h00
[7:6]	Reserved	Reserved	RO	-
[5:0]	scg#{index*2}_top_address_bit0	Top address bit 0 for system cache group #{index*2}	RW	6'h00

4.3.16.47 sam_qos_mem_region_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the QoS value for memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`index(0-15) : 16'h1280 + #{8 * index}`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-637: sam_qos_mem_region_reg0-15

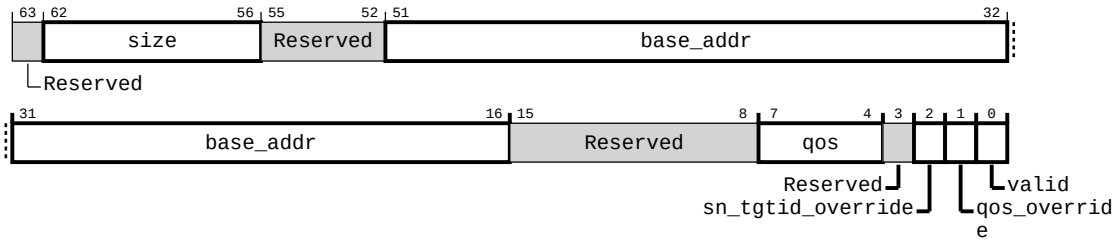


Table 4-653: sam_qos_mem_region_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	size	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	36'h0
[15:8]	Reserved	Reserved	RO	-
[7:4]	qos	Indicates the QoS value to be used for this region	RW	4'b0000
[3]	Reserved	Reserved	RO	-
[2]	sn_tgtid_override	Override the SN targetId for address contained in the region of this register	RW	1'b0
[1]	qos_override	QoS Memory region allow override **1'b0** Do not override the QoS value from the QoS regulator **1'b1** Override the QoS value with the programmed value in regionX_qos	RW	1'b0

Bits	Name	Description	Type	Reset
[0]	valid	QoS Memory region valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.48 sam_qos_mem_region_cfg2_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the QOS memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-15) : 16'h1340 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-638: sam_qos_mem_region_cfg2_reg0-15

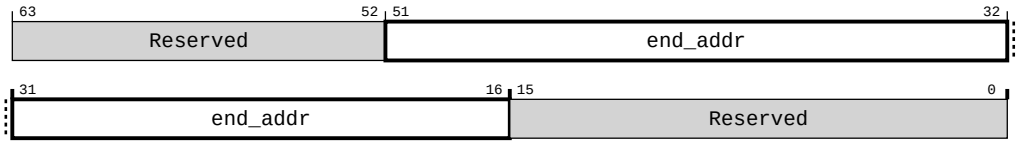


Table 4-654: sam_qos_mem_region_cfg2_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	36'h0
[15:0]	Reserved	Reserved	RO	-

4.3.16.49 sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg1_reg0-511%64

There are 512 iterations of this register. The index ranges from 0 to 511. Configures the prefetch nonhash memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-511) : 16'h4000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-639: sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg1_reg0-511%64

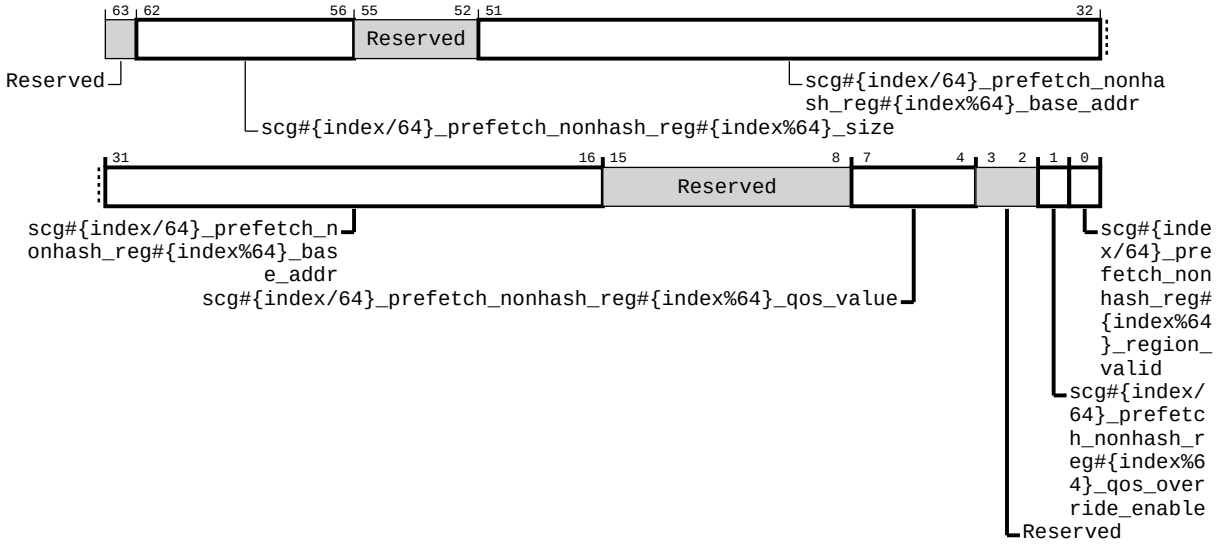


Table 4-655: sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg1_reg0-511%64 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	scg#{index/64}_prefetch_nonhash_reg#{index%64}_size	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	scg#{index/64}_prefetch_nonhash_reg#{index%64}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	36'h0
[15:8]	Reserved	Reserved	RO	-
[7:4]	scg#{index/64}_prefetch_nonhash_reg#{index%64}_qos_value	Indicates the QoS value to be used for this region	RW	4'b0000
[3:2]	Reserved	Reserved	RO	-
[1]	scg#{index/64}_prefetch_nonhash_reg#{index%64}_qos_override_enable	Prefetch nonhash allow QOS override **1'b0** Do not override the QoS value from the QoS regulator **1'b1** Override the QoS value with the programmed value in regionX_qos	RW	1'b0
[0]	scg#{index/64}_prefetch_nonhash_reg#{index%64}_region_valid	Prefetch Nonhash region valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.50 sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg2_reg0-511%64

There are 512 iterations of this register. The index ranges from 0 to 511. Configures the Prefetch nonhash memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-511) : 16'h5000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-640: sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg2_reg0-511%64

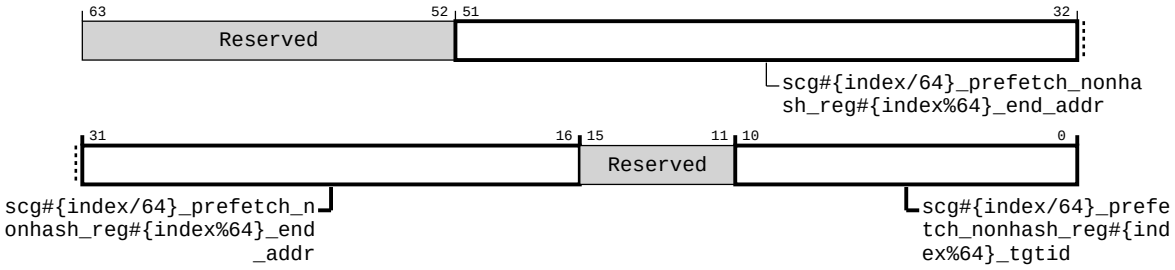


Table 4-656: sam_scg0-511/64_prefetch_nonhashed_mem_region_cfg2_reg0-511%64 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	scg#{index/64}_prefetch_nonhash_reg#{index%64}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	36'h0

Bits	Name	Description	Type	Reset
[15:11]	Reserved	Reserved	RO	-
[10:0]	scg#{index/64}_prefetch_nonhash_reg#{index %64}_tgtid	SN TgtID for the non-hashed region	RW	11'h0

4.3.16.51 sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8

There are 64 iterations of this register. The index ranges from 0 to 63. Configures the prefetch hashed memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-63) : 16'h6000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-641: sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8

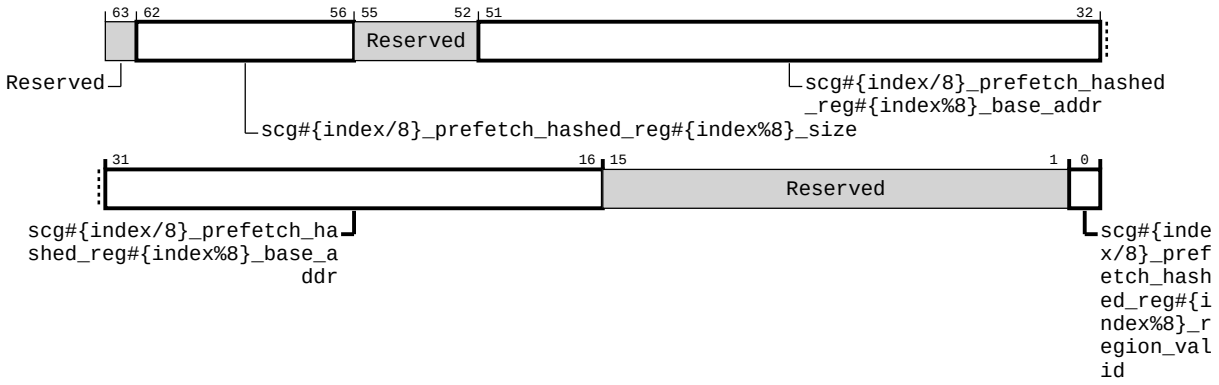


Table 4-657: sam_scg0-63/8_prefetch_hashed_region_cfg1_reg0-63%8 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	scg#{index/8}_prefetch_hashed_reg#{index%8}_size	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	scg#{index/8}_prefetch_hashed_reg#{index%8}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	36'h0
[15:1]	Reserved	Reserved	RO	-
[0]	scg#{index/8}_prefetch_hashed_reg#{index%8}_region_valid	Prefetch Hashed region valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.52 sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8

There are 64 iterations of this register. The index ranges from 0 to 63. Configures the Prefetch hashed memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-63) : 16'h6200 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-642: sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8

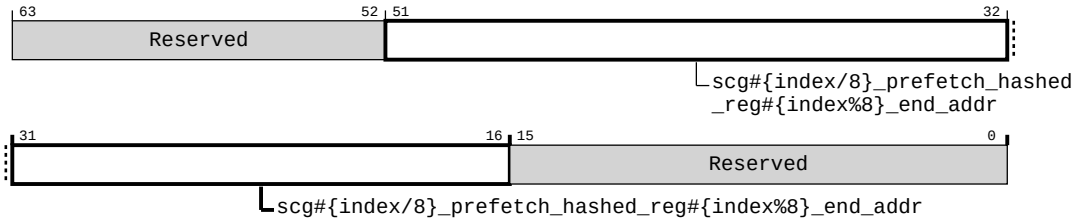


Table 4-658: sam_scg0-63/8_prefetch_hashed_region_cfg2_reg0-63%8 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	scg#{index/8}_prefetch_hashed_reg#{index%8}_end_addr	Bits [51:16] of end address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	36'h0
[15:0]	Reserved	Reserved	RO	-

4.3.16.53 sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8

There are 64 iterations of this register. The index ranges from 0 to 63. Configures the Prefetch hashed memory region #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-63) : 16'h6400 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-643: sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8

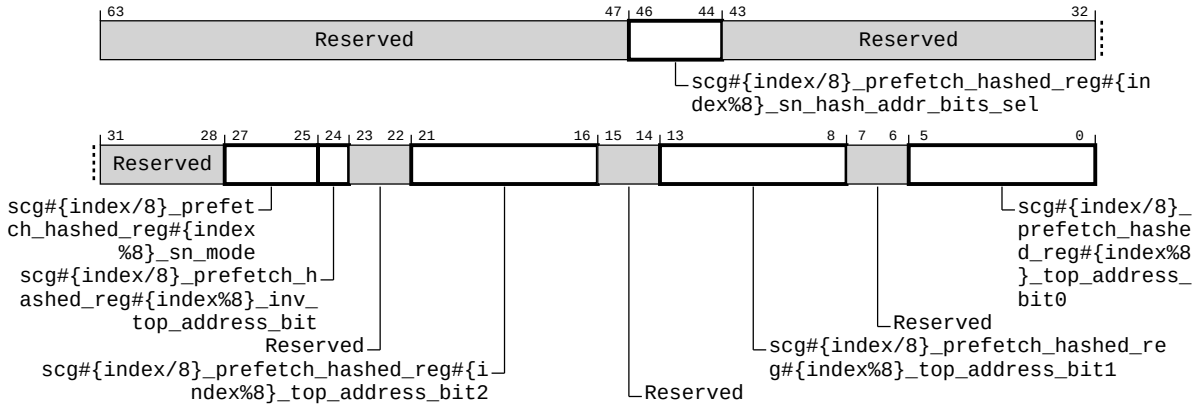


Table 4-659: sam_scg0-63/8_prefetch_hashed_region_cfg3_reg0-63%8 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[46:44]	scg#{index/8}_prefetch_hashed_reg#{index%8}_sn_hash_addr_bits_sel	SN hash address select(Valid for 3SN, 5SN, 6SN) **3'b000** [16:8] address bits (Default) **3'b001** [17:9] address bits **3'b010** [18:10] address bits **3'b011** [19:11] address bits **3'b100** [20:12] address bits **3'b101** [21:13] address bits **Others** Reserved	RW	3'h0
[43:28]	Reserved	Reserved	RO	-
[27:25]	scg#{index/8}_prefetch_hashed_reg#{index%8}_sn_mode	SN selection mode 3'b000: Reserved 3'b001: 3-SN mode (SN0, SN1, SN2) 3'b010: 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) 3'b011: 5-SN mode (SN0, SN1, SN2, SN3, SN4) 3'b100: 2-SN mode (SN0, SN1) power of 2 hashing 3'b101: 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing 3'b110: 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing 3'b111: Reserved	RW	3'b0
[24]	scg#{index/8}_prefetch_hashed_reg#{index%8}_inv_top_address_bit	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	scg#{index/8}_prefetch_hashed_reg#{index%8}_top_address_bit2	Top address bit 2	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	scg#{index/8}_prefetch_hashed_reg#{index%8}_top_address_bit1	Top address bit 1	RW	6'h00
[7:6]	Reserved	Reserved	RO	-
[5:0]	scg#{index/8}_prefetch_hashed_reg#{index%8}_top_address_bit0	Top address bit 0	RW	6'h00

4.3.16.54 sys_cache_grp_sn_nodeid_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed node IDs for system cache groups. Controls target SN node IDs #{index*4} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h1000 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-644: sys_cache_grp_sn_nodeid_reg0-31

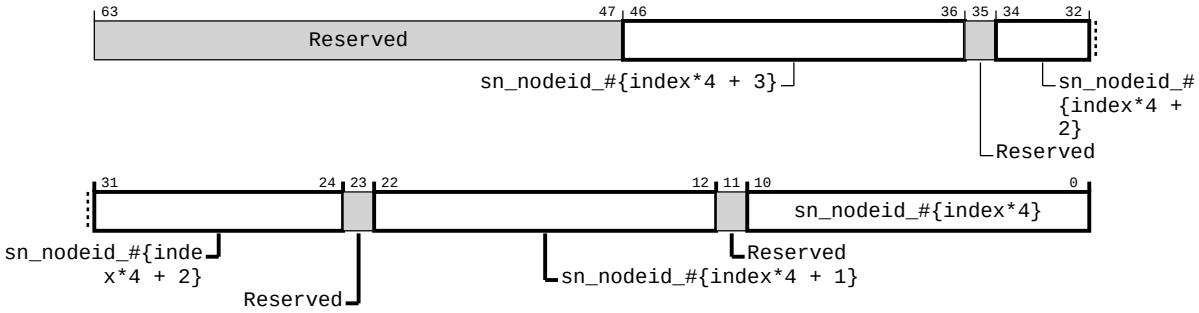


Table 4-660: sys_cache_grp_sn_nodeid_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	sn_nodeid_#{index*4 + 3}	Default Hashed target SN node ID #{index*4 + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	sn_nodeid_#{index*4 + 2}	Default Hashed target SN node ID #{index*4 + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	sn_nodeid_#{index*4 + 1}	Default Hashed target SN node ID #{index*4 + 1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_#{index*4}	Default Hashed target SN node ID #{index*4}	RW	11'b00000000000

4.3.16.55 sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32

There are 64 iterations of this register. Configures node IDs for SCG's Default hashed Region memory. Controls target SN node IDs #{index*4} to #{index*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-63) : 16'h1400 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-645: sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32

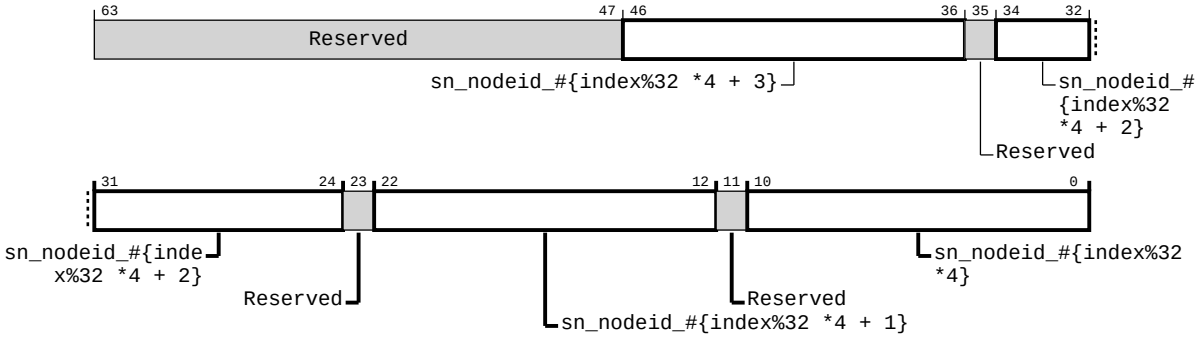


Table 4-661: sys_cache_grp_region0-63/32_sn_nodeid_reg0-63%32 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	sn_nodeid_#{index%32 *4 + 3}	Hashed target SN node ID #{index%32 * 4 +3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	sn_nodeid_#{index%32 *4 + 2}	Hashed target SN node ID #{index%32 * 4 +2}	RW	11'b00000000000

Bits	Name	Description	Type	Reset
[23]	Reserved	Reserved	RO	-
[22:12]	sn_nodeid_#{index%32 *4 + 1}	Hashed target SN node ID #{index%32 * 4 +1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_#{index%32 *4}	Hashed target SN node ID #{index%32 * 4}	RW	11'b00000000000

4.3.16.56 sys_cache_grp_hashed_regions_sn_nodeid_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures SN node IDs for SCG's Hashed groups in the HNSAM . Controls target SN node IDs #{index*4} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-31) : 16'h6600 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-646: sys_cache_grp_hashed_regions_sn_nodeid_reg0-15

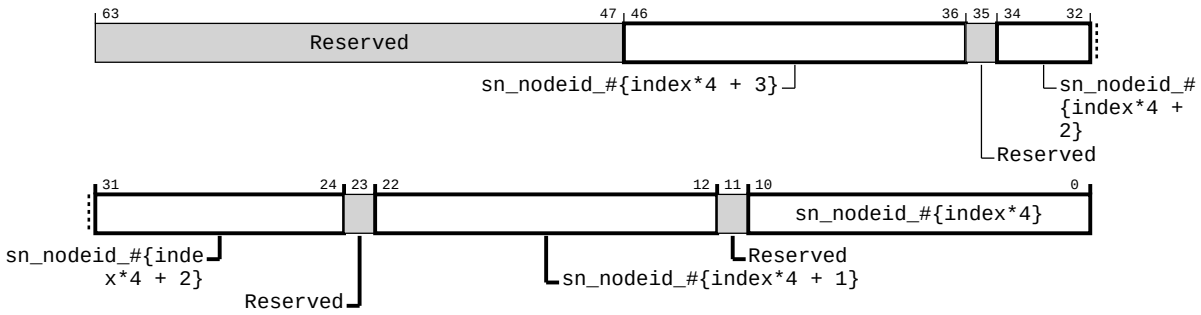


Table 4-662: sys_cache_grp_hashed_regions_sn_nodeid_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	sn_nodeid_#{index*4 + 3}	Hashed target SN node ID #{index*4 + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	sn_nodeid_#{index*4 + 2}	Hashed target SN node ID #{index*4 + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	sn_nodeid_#{index*4 + 1}	Hashed target SN node ID #{index*4 + 1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_#{index*4}	Hashed target SN node ID #{index*4}	RW	11'b00000000000

4.3.16.57 rnsam_status

Functions as the default and programming mode status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1100`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-647: rnsam_status

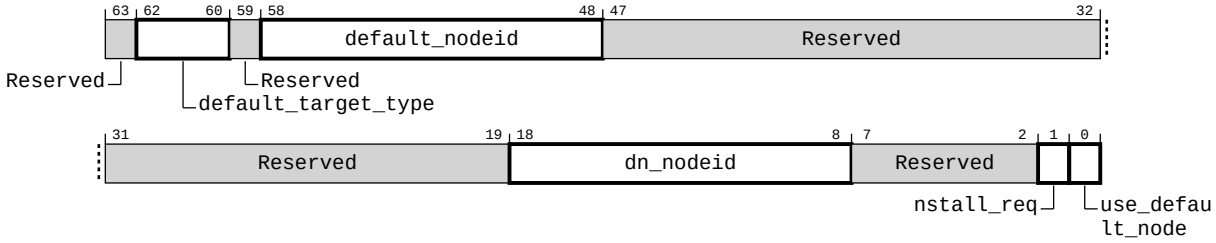


Table 4-663: rnsam_status attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:60]	default_target_type	Indicates node type **3'b000** HN-F **3'b001** HN-I **3'b010** CXRA **3'b011** HN-P **3'b100** PCI-CXRA **3'b101** HN-S **Others** Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b001
[59]	Reserved	Reserved	RO	-
[58:48]	default_nodeid	Default Node ID	RW	Configuration dependent
[47:19]	Reserved	Reserved	RO	-
[18:8]	dn_nodeid	DN Node ID for DN operations	RW	11'b0
[7:2]	Reserved	Reserved	RO	-
[1]	nstall_req	Indicates RN SAM is programmed and ready **1'b0** STALL requests **1'b1** UNSTALL requests	RW	1'b0
[0]	use_default_node	Indicates target ID selection mode **1'b0** Enables RN SAM to hash address bits and generate target ID **1'b1** Uses default target ID	RW	1'b1

4.3.16.58 gic_mem_region_reg

Configures GIC memory region.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h1108`

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-648: gic_mem_region_reg

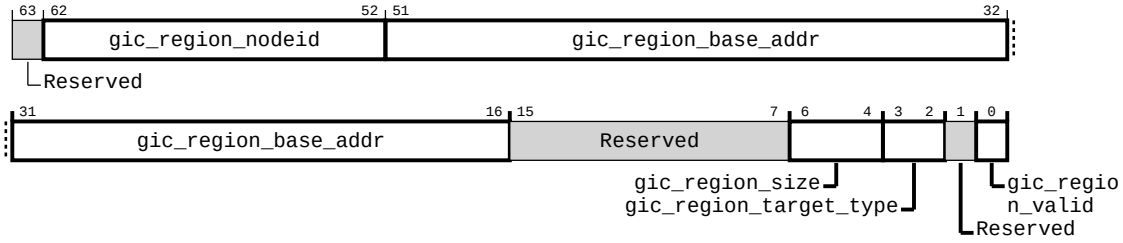


Table 4-664: gic_mem_region_reg attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:52]	gic_region_nodeid	GIC node ID	RW	11'b00000000000
[51:16]	gic_region_base_addr	Base address of the GIC memory region CONSTRAINT: Must be an integer multiple of region size	RW	36'h000000000
[15:7]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[6:4]	gic_region_size	GIC memory region size **3'b000** 64KB **3'b001** 128KB **3'b010** 256KB **3'b011** 512KB CONSTRAINT: Memory region must be a power of 2.	RW	3'b000
[3:2]	gic_region_target_type	Indicates node type **2'b00** HN-F **2'b01** HN-I **2'b10** CXRA **2'b11** HN-P CONSTRAINT: Only applicable for RN-I	RW	2'b00
[1]	Reserved	Reserved	RO	-
[0]	gic_region_valid	Memory region 1 valid **1'b0** not valid **1'b1** valid for memory region comparison	RW	1'b0

4.3.16.59 sam_generic_regs0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configuration register for the custom logic

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

```
index(0-7) : 16'h1600 + #{8 * index}
```

Type

RW

Reset value

See individual bit resets

Secure group override

por_rnsam_secure_register_groups_override.mem_range

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-649: sam_generic_regs0-7

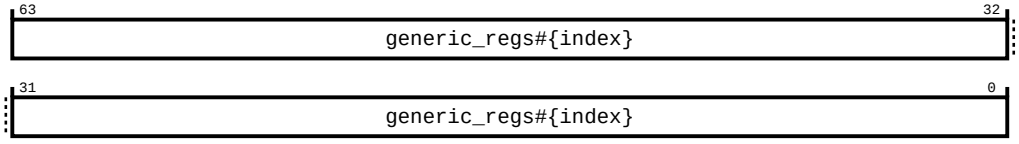


Table 4-665: sam_generic_regs0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	generic_regs#{index}	Configuration register for the custom logic	RW	64'h0

4.3.17 SBSX register descriptions

This section lists the SBSX registers.

4.3.17.1 por_sbsx_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h0`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-650: por_sbsx_node_info

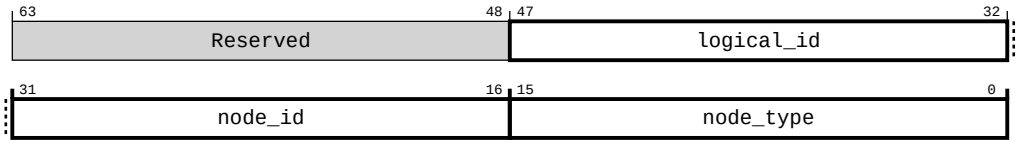


Table 4-666: por_sbsx_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	node_id	Component node ID	RO	Configuration dependent
[15:0]	node_type	CMN-700 node type identifier	RO	16'h0007

4.3.17.2 por_sbsx_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h80`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-651: por_sbsx_child_info

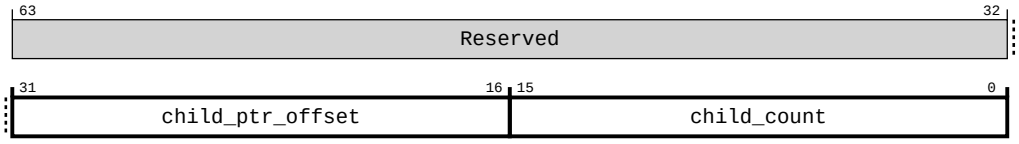


Table 4-667: por_sbsx_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

4.3.17.3 por_sbsx_secure_register_groups_override

Allows Non-secure access to predefined groups of Secure registers.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h980`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-652: por_sbsx_secure_register_groups_override

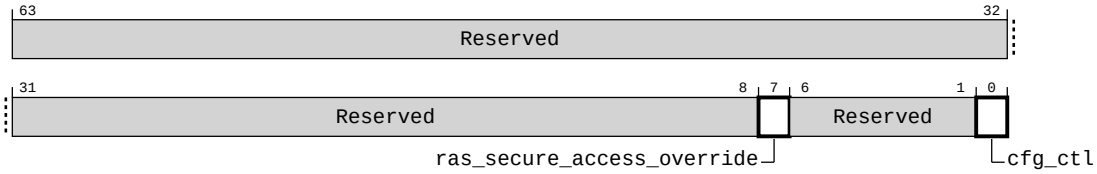


Table 4-668: por_sbsx_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7]	ras_secure_access_override	Allow Non-secure access to Secure RAS registers	RW	1'b0
[6:1]	Reserved	Reserved	RO	-
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register (por_sbsx_cfg_ctl)	RW	1'b0

4.3.17.4 por_sbsx_unit_info

Provides component identification information for SBSX.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h900`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-653: por_sbsx_unit_info

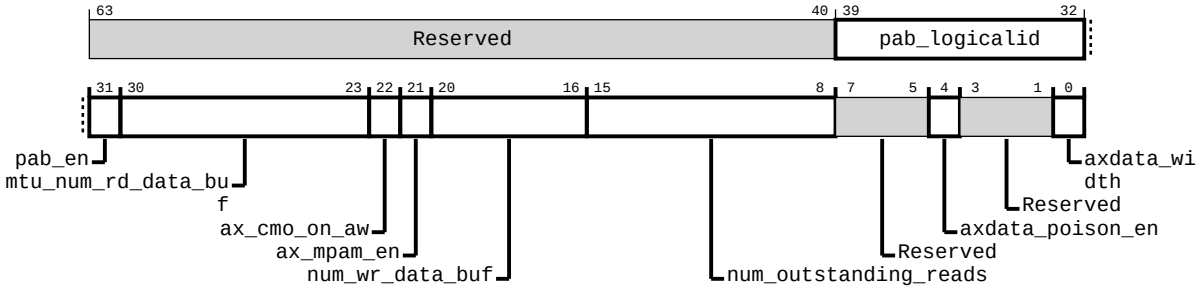


Table 4-669: por_sbsx_unit_info attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	pab_logicalid	PUB AUB bridge Logical ID	RO	Configuration dependent
[31]	pab_en	PUB AUB bridge enable **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[30:23]	mtu_num_rd_data_buf	Number of mtu read data buffers in SBSX	RO	Configuration dependent
[22]	ax_cmo_on_aw	Write Channel CMOs enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[21]	ax_mpam_en	MPAM enable on ACE-Lite/AXI4 interface **1'b1** Enabled **1'b0** Not enabled	RO	Configuration dependent
[20:16]	num_wr_data_buf	Number of write data buffers in SBSX	RO	Configuration dependent
[15:8]	num_outstanding_reads	Maximum number of outstanding AXI read requests from SBSX	RO	Configuration dependent
[7:5]	Reserved	Reserved	RO	-
[4]	axdata_poison_en	Data poison support on ACE-Lite/AXI4 interface **1'b0** Not supported **1'b1** Supported	RO	Configuration dependent
[3:1]	Reserved	Reserved	RO	-
[0]	axdata_width	Data width on ACE-Lite/AXI4 interface **1'b0** 128 bits **1'b1** 256 bits	RO	Configuration dependent

4.3.17.5 por_sbsx_cfg_ctl

Functions as the configuration control register for SBSX bridge.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA00`

Type

RW

Reset value

See individual bit resets

Secure group override

por_sbsx_secure_register_groups_override.cfg_ctl

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-654: por_sbsx_cfg_ctl

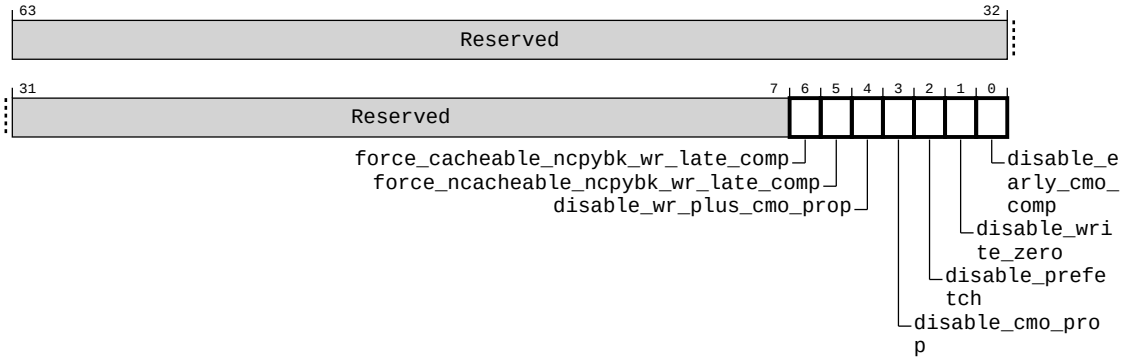


Table 4-670: por_sbsx_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[6]	force_cacheable_ncpybk_wr_late_comp	Late Comp for Cacheable Non-CopyBack Writes. Overrides EWA	RW	1'b0

Bits	Name	Description	Type	Reset
[5]	force_ncacheable_ncpybk_wr_late_comp	Late Comp for Non-cacheable Non-CopyBack Writes. Overrides EWA	RW	1'b0
[4]	disable_wr_plus_cmo_prop	Disables Write_plus_CMO propagation on ACE.	RW	1'b0
[3]	disable_cmo_prop	Disables CMO propagation on ACE.	RW	1'b0
[2]	disable_prefetch	Disables Prefetches on AXI.	RW	1'b0
[1]	disable_write_zero	Disables WriteZero Op on AXI.	RW	1'b0
[0]	disable_early_cmo_comp	Disables Early Comp for CMOs in SBSX to HNF.	RW	1'b0

4.3.17.6 por_sbsx_aux_ctl

Functions as the auxiliary control register for the SBSX bridge.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA08`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-655: por_sbsx_aux_ctl

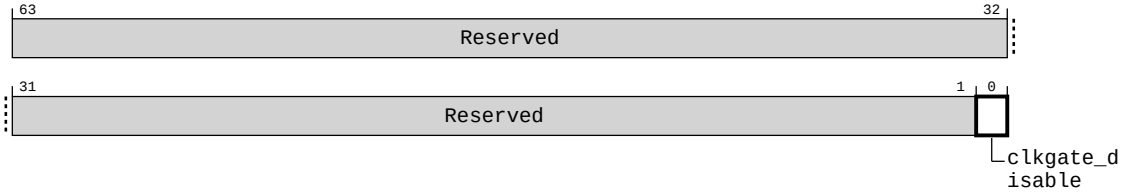


Table 4-671: por_sbsx_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	clkgate_disable	Disables internal clock gating in SBSX bridge	RW	1'b0

4.3.17.7 por_sbsx_cbusy_limit_ctl

Cbusy threshold limits for Request Tracker entries.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'hA18`

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-656: por_sbsx_cbusy_limit_ctl

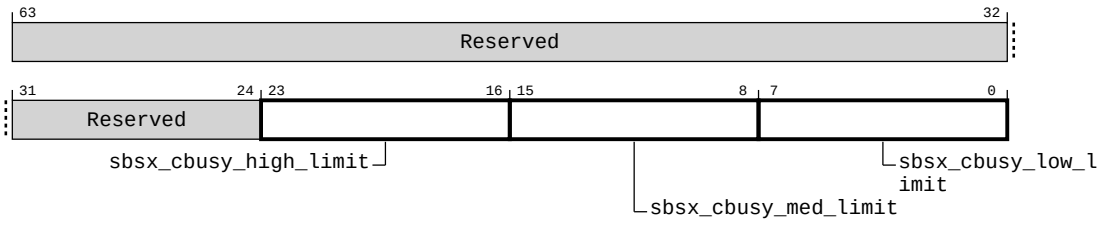


Table 4-672: por_sbsx_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	sbsx_cbusy_high_limit	ReqTracker limit for CBusy High	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[15:8]	sbsx_cbusy_med_limit	ReqTracker limit for CBusy Med	RW	Configuration dependent
[7:0]	sbsx_cbusy_low_limit	ReqTracker limit for CBusy Low	RW	Configuration dependent

4.3.17.8 por_sbsx_errfr

Functions as the error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3000`

Type

RO

Reset value

See individual bit resets

Secure group override

por_sbsx_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-657: por_sbsx_errfr

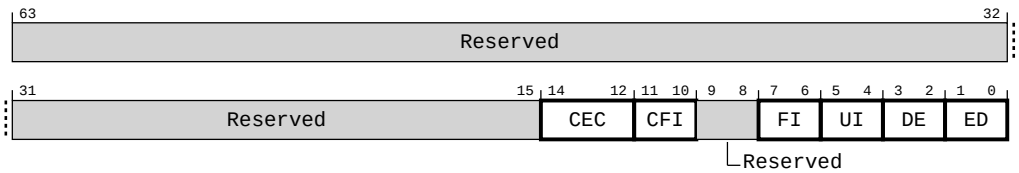


Table 4-673: por_sbsx_errfr attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism 3'b000: Does not implement standardized error counter model	RO	3'b000

Bits	Name	Description	Type	Reset
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors	RO	2'b00
[1:0]	ED	Error detection	RO	2'b01

4.3.17.9 por_sbsx_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3008`

Type

RW

Reset value

See individual bit resets

Secure group override

por_sbsx_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-658: por_sbsx_errctlr

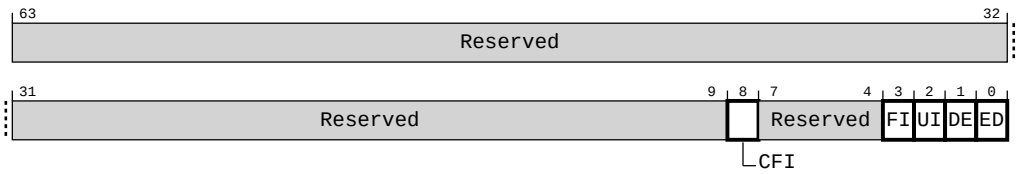


Table 4-674: por_sbsx_errctlr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_sbsx_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_sbsx_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_sbsx_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_sbsx_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_sbsx_errfr.ED	RW	1'b0

4.3.17.10 por_sbsx_errstatus

Functions as the error status register. AV and MV bits must be cleared in the same cycle, otherwise the error record does not have a consistent view.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3010`

Type

W1C

Reset value

See individual bit resets

Secure group override

por_sbsx_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-659: por_sbsx_errstatus

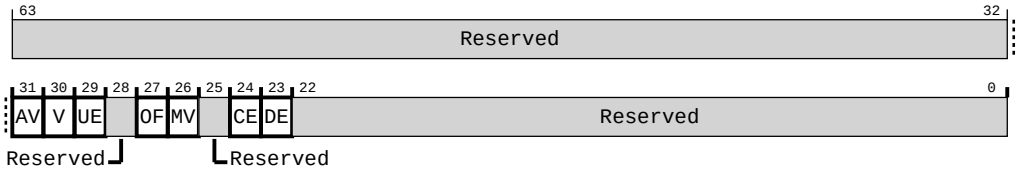


Table 4-675: por_sbsx_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; por_sbsx_erraddr contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	por_sbsx_errmisc valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.17.11 por_sbsx_erraddr

Contains the error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3018`

Type

RW

Reset value

See individual bit resets

Secure group override

por_sbsx_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-660: por_sbsx_erraddr

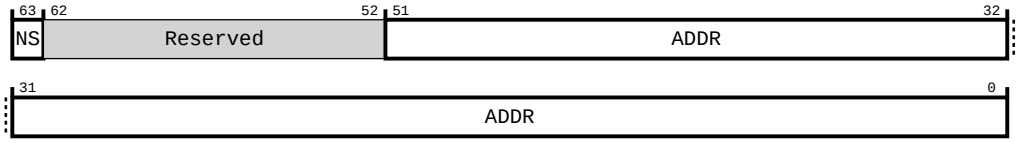


Table 4-676: por_sbsx_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: por_sbsx_erraddr.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.17.12 por_sbsx_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3020`

Type

RW

Reset value

See individual bit resets

Secure group override

por_sbsx_secure_register_groups_override.ras_secure_access_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-661: por_sbsx_errmisc

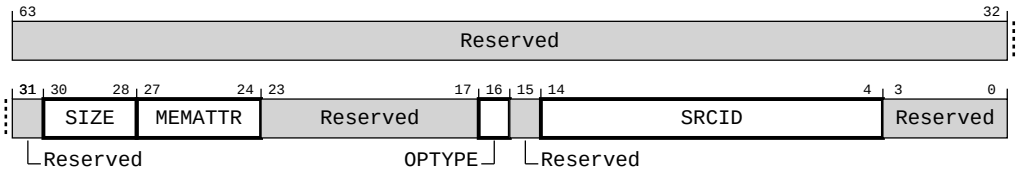


Table 4-677: por_sbsx_errmisc attributes

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-
[30:28]	SIZE	Error transaction size	RW	3'b0
[27:24]	MEMATTR	Error memory attributes	RW	4'b0
[23:17]	Reserved	Reserved	RO	-
[16]	OPTYPE	Error opcode type **1'b1** WR_NO_SNP_PTL (partial) **1'b0** WR_NO_SNP_FULL	RW	1'b0
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0
[3:0]	Reserved	Reserved	RO	-

4.3.17.13 por_sbsx_errfr_NS

Functions as the Non-secure error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3100`

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-662: por_sbsx_errfr_NS

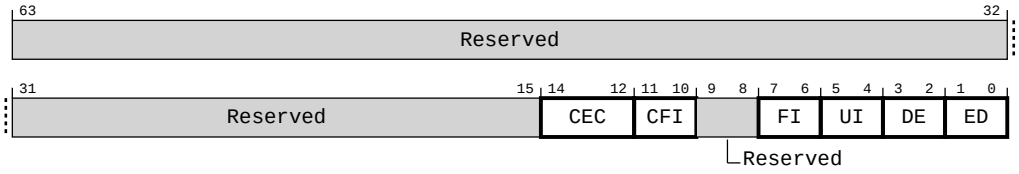


Table 4-678: por_sbsx_errfr_NS attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:12]	CEC	Standard corrected error count mechanism 3'b000: Does not implement standardized error counter model	RO	3'b000
[11:10]	CFI	Corrected error interrupt	RO	2'b00
[9:8]	Reserved	Reserved	RO	-
[7:6]	FI	Fault handling interrupt	RO	2'b10
[5:4]	UI	Uncorrected error interrupt	RO	2'b10
[3:2]	DE	Deferred errors	RO	2'b00
[1:0]	ED	Error detection	RO	2'b01

4.3.17.14 por_sbsx_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3108`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-663: por_sbsx_errctlr_NS

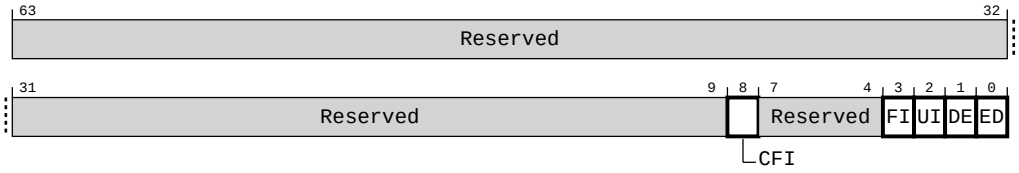


Table 4-679: por_sbsx_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_sbsx_errfr_NS.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for all detected deferred errors as specified in por_sbsx_errfr_NS.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_sbsx_errfr_NS.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_sbsx_errfr_NS.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_sbsx_errfr_NS.ED	RW	1'b0

4.3.17.15 por_sbsx_errstatus_NS

Functions as the Non-secure error status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3110`

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-664: por_sbsx_errstatus_NS

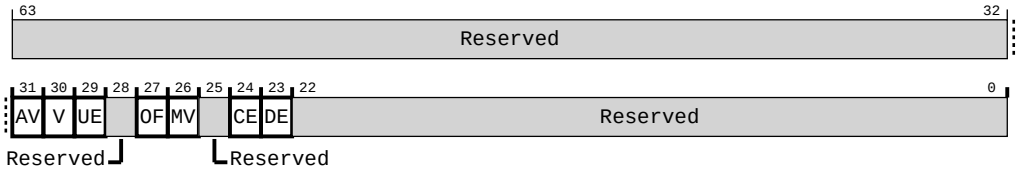


Table 4-680: por_sbsx_errstatus_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Address is valid; por_sbsx_erraddr_NS contains a physical address for that recorded error **1'b0** Address is not valid	W1C	1'b0
[30]	V	Register valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and are not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error recorded; register is valid **1'b0** No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error detected that is not corrected and is not deferred to a subordinate **1'b0** No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow; asserted when multiple errors of the highest priority type are detected; write a 1 to clear **1'b1** More than one error detected **1'b0** Only one error of the highest priority type detected as described by UE/DE/CE fields	W1C	1'b0

Bits	Name	Description	Type	Reset
[26]	MV	por_sbsx_errmisc_NS valid; writes to this bit are ignored if any of the UE, DE, or CE bits are set to 1, and the highest priority are not cleared to 0 in the same write; write a 1 to clear **1'b1** Miscellaneous registers are valid **1'b0** Miscellaneous registers are not valid	W1C	1'b0
[25]	Reserved	Reserved	RO	-
[24]	CE	Corrected errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one transient corrected error recorded **1'b0** No corrected errors recorded	W1C	1'b0
[23]	DE	Deferred errors; writes to this bit are ignored if the OF bit is set to 1, and is not cleared to 0 in the same write; write a 1 to clear **1'b1** At least one error is not corrected and is deferred **1'b0** No errors deferred	W1C	1'b0
[22:0]	Reserved	Reserved	RO	-

4.3.17.16 por_sbsx_erraddr_NS

Contains the Non-secure error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3118`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-665: por_sbsx_erraddr_NS

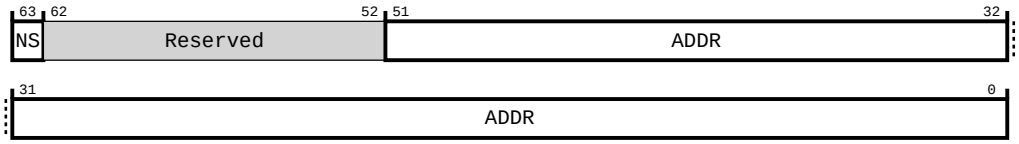


Table 4-681: por_sbsx_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction **1'b1** Non-secure transaction **1'b0** Secure transaction CONSTRAINT: por_sbsx_erraddr_NS.NS is redundant. Since it is writable, it cannot be used for logic qualification.	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

4.3.17.17 por_sbsx_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred/uncorrected errors.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h3120`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-666: por_sbsx_errmisc_NS

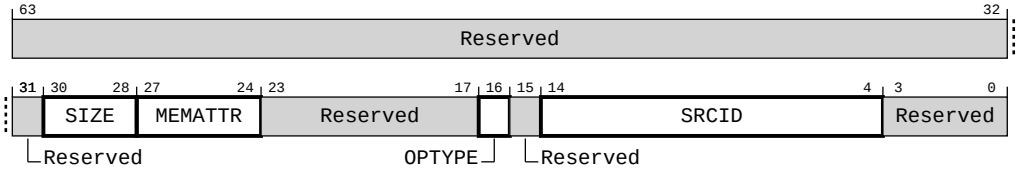


Table 4-682: por_sbsx_errmisc_NS attributes

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-
[30:28]	SIZE	Error transaction size	RW	3'b0
[27:24]	MEMATTR	Error memory attributes	RW	4'b0
[23:17]	Reserved	Reserved	RO	-
[16]	OPTYPE	Error opcode type **1'b1** WR_NO_SNP_PTL (partial) **1'b0** WR_NO_SNP_FULL	RW	1'b0
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0
[3:0]	Reserved	Reserved	RO	-

4.3.17.18 por_sbsx_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

`16'h2000`

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 4-667: por_sbsx_pmu_event_sel

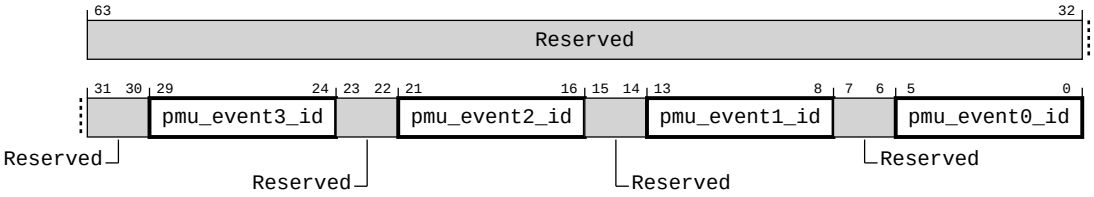


Table 4-683: por_sbsx_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	SBSX PMU Event 3 select; see pmu_event0_id for encodings	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	SBSX PMU Event 2 select; see pmu_event0_id for encodings	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	SBSX PMU Event 1 select; see pmu_event0_id for encodings	RW	6'b0
[7:6]	Reserved	Reserved	RO	-
[5:0]	pmu_event0_id	SBSX PMU Event 0 select **6'h00** No event **6'h01** Read request **6'h02** Write request **6'h03** CMO request **6'h04** RETRYACK TXRSP flit sent **6'h05** TXDAT flit seen **6'h06** TXRSP flit seen **6'h11** Read request tracker occupancy count overflow **6'h12** Write request tracker occupancy count overflow **6'h13** CMO request tracker occupancy count overflow **6'h14** WDB occupancy count overflow **6'h15** Read AXI pending tracker occupancy count overflow **6'h16** CMO AXI pending tracker occupancy count overflow **6'h17** RDB occupancy count overflow. (Only when MTU is enabled) **6'h21** ARVALID set without ARREADY **6'h22** AWVALID set without AWREADY **6'h23** WVALID set without WREADY **6'h24** TXDAT stall (TXDAT valid but no link credit available) **6'h25** TXRSP stall (TXRSP valid but no link credit available) NOTE: All other encodings are reserved.	RW	6'b0

4.4 CMN-700 programming

This section contains CMN-700 programming information.

4.4.1 Boot-time programming sequence

A specific boot-time programming sequence must be used to set up CMN-700 correctly. An example sequence is provided, which uses a *System Control Processor* (SCP) to perform the initial boot configuration.

After reset, the following configuration steps must happen before broad access to CMN-700 components is available:

1. CMN-700 uses a default configuration to access boot flash through the HN-D ACE-Lite manager interface and also the configuration registers.
2. An RN-F, or a manager that is connected to an RN-I, must then access the configuration registers to configure CMN-700. This boot-time configuration must happen before there is broader access to components such as HN-F or SN.


Note

If booting from a device downstream of the HN-D node, in your final RN SAM configuration the boot code region target must not change from the HN-D.

The following example provides more information on the boot process. It assumes an SCP is performing the CMN-700 configuration.

1. The SCP boots, either from local memory or through CMN-700 memory accesses targeting memory behind the HN-D:
   - All other managers are either held in reset or issue no requests to CMN-700 until the boot programming is complete.
   - The HN-D is identified through straps on the RN SAM.
2. If necessary, the SCP discovers the system.
3. The SCP determines the wanted address map and corresponding SAM register values.
4. If necessary, the SCP remaps the configuration register space by completing the following steps:
   a. It drains all requests in flight by waiting for their responses.
   b. It issues a single 64-bit store to a PERIPHBASE register behind the HN-D. This register would be in logic that is external to CMN-700 and an update would cause the signal values on the CFGM_PERIPHBASE input to change.
   c. It waits for the response for that store.
5. If necessary, the SCP writes to the CMN-700 configuration registers to program the SAM for all HN-Fs.

6. The SCP writes to the CMN-700 configuration registers to program the SAM for all RNs including the one being used by the SCP.


Note

RN-F ESAM types are not able to block transactions before RN SAM programming and some external mechanism must block transactions. CMN-700 only supports RN-F ESAM types.

After programming the SAM for all RNs, the SCP sets a bit that enables use of the programmed address map instead of the default address map. This bit indicates that the SAM setup is complete.

When the preceding steps are complete, the SCP can make general accesses anywhere in the address space and other managers can begin issuing requests.

4.4.2 Runtime programming requirements

This section describes the requirements for programming during runtime.

The hardware for handling RN membership in the coherence domain or DVM domain has been shifted to the XP to which the RN is attached. A low-level four-phase handshake mechanism (SYSCOREQ/SYSCOACK) has been added to allow quick and local entry to and exit from snoop and DVM domains. No communication with central hardware resources is required. When a block is removed from the coherence or DVM domain, the XP acts as a protocol agent to give a generic response to any snoop or DVM messages.

For legacy devices that do not support the SYSCOREQ/SYSCOACK mechanism, direct configuration writes to the XP by software can trigger the same mechanism. See 3.3.8 RN entry to and exit from Snoop and DVM domains on page 98.

4.4.3 RN SAM and HN-F SAM programming

You must follow specific programming sequences to set up the RN SAM and HN-F SAM correctly. The register operating modes and encodings you use depend on your system configuration and requirements.

4.4.3.1 Program the SAM

The SAM must be programmed using a specific sequence. An RN-F or manager that is connected to an RN-I must perform this sequence during the configuration of CMN-700 at boot.

Before you begin

This sequence is part of the overall CMN-700 boot configuration process. There are steps that must occur at boot-time before SAM programming. For more information about the full process, see 4.4.1 Boot-time programming sequence on page 1120.

All MBISTREQ and nMBISTRESET signals must be disabled during functional operation. The P-Channel, Q-Channel, ACLKEN_M, and ACLKEN_S signals must also all be set correctly for SAM programming.

About this task

There are several configuration decisions that must be made when setting up the SAM. See the following sections:

- 3.4.6 RN SAM on page 112
- 3.4.7 HN-F SAM on page 147

Procedure

1. Define the following memory map regions:
   - Hashed memory regions, which target HN-Fs (address based hashing), HN-P's (AXID based hashing). The hashed memory regions can be partitioned into SCG's or HTG's, if applicable.
   - Non-hashed regions with HN-I, HN-D, or HN-P mapping. If a single HN-F is the target, HN-F can also be used in non-hashed mode.
   - GIC memory region, if present.
   - HN-F SAM memory regions, if applicable.
   - Mapping of HN-Fs to SN-Fs. This mapping can be direct, 3-SN, 5-SN, or 6-SN mode.
2. Ensure that the memory map meets the following requirements:
   - Non-hashed memory regions must not overlap.
   - Hashed memory regions must not overlap.
   - The memory regions must be size-aligned (if the region compare parameters are not enabled).
3. Program the following attributes and registers for each HN-F SAM:
   a) Program the appropriate properties for each SN-F ID, according to the features that are supported in the cmn_hns_sam_sn_properties register.
   These properties provide the interface width as either 128-bit or 256-bit, CMO support, and PCMO support.
   b) Program the HN-F to SN-F mapping, which depends on the mapping schemes that are used:
      - If the HN-F is directly mapped to an SN-F, program the SN0 target ID and corresponding attributes.
      - If the HN-F is in 3-SN, 5-SN, or 6-SN mode, program the following:
        - All SN-F target IDs and the attributes for each SN-F.
        - The mode of operation as 3-SN, 5-SN, or 6-SN.
        - The top address bits.
      - If the HN-F uses range-based SN-F partitioning for a particular memory region, program the memory region registers, including the target ID that is associated with each region.
4. Complete the following programming for the RN-F RN SAM:
   a) Program the following attributes and registers for each SCG:

- Memory region registers.
- HN-F count registers.
- HN-F target ID registers.
- If PrefetchTarget operations are enabled:
  - SN-F target ID registers for SCG.
  - SN-F target ID selection mode for SCG.
  - If 3-SN, 5-SN, or 6-SN mode is enabled, program the top address bits.

b) Program the non-hashed memory region registers.
c) Program the non-hashed target ID registers.
d) Program the rnsam_status register to disable the default target ID mode.

5. Complete the following programming for each RN-I RN SAM and RN-D RN SAM:
   a) Program the following registers for each SCG:
      - Memory region registers.
      - HN-F count registers.
      - HN-F target ID registers.
   b) Program the non-hashed memory region registers.
   c) Program the non-hashed target ID registers.
   d) Program the rnsam_status register to disable the default target ID mode.

4.4.3.2 SAM programming examples

This section contains the programming examples of the different devices supported by the CMN-700

4.4.3.2.1 RNSAM programming with power-of-two hashing

This is an example for a system with 32-HNFs, without CAL, HNSAM uses 3-SN or Direct-mapping mode.

Hashing function:

Select[4:0] = { (10^15^20^...^50),

(9^14^19^...^49),

(8^13^18^...^48),

(7^12^17^...^47),

(6^11^16^...^51) }

Example 4-43: RNSAM programming

```
##################
```

```
# Set up SCG0
##################
sys_cache_grp_region0.region0_valid = 1'b1;             # Set SCG0 Valid
sys_cache_grp_region0.region0_target_type = 3'b000;     # target type HNF
sys_cache_grp_region0.region0_base_addr = <base address for SCG>;
                                                        # SCG0 Base Address
sys_cache_grp_region0.region0_size = <size of the SCG>;      # SCG0 size
sys_cache_group_hn_count.scg0_num_hnf = 8'h20;       # 32 HNs in SCG0

##################
# Program the 32 HNF targetIDs -
# sys_cache_grp_hn_nodeid_reg[7:0]  covers 32 HNF targetIDs for cluster0
##################
for (indx=0, indx=indx+1, indx <8) begin    # 8 regs for 32 HNF nodeIDs
 for (id=0, id=id+1, id <4) begin          # each reg covers 4 nodeIDs
  sys_cache_grp_hn_nodeid_reg#{indx}.nodeid_#{4*indx + id} = <HNF#{4*indx +
 id}targetID>
   end
end

#HNSAM programming
##################
# 3-SN mode
##################

#Program each HNF to 3-SN mode
cmn_hns_sam_control.hn_cfg_three_sn_en = 1'b1;
cmn_hns_sam_6sn_nodeid.hn_hash_addr_bits_sel = 3'b001;
cmn_hns_sam_control.hn_cfg_sam_top_address_bit0 = 6'd32;
                                    # depends on customer memory map
cmn_hns_sam_control.hn_cfg_sam_top_address_bit1 = 6'd33;
                                    # depends on customer memory map

cmn_hns_sam_control.hn_cfg_sn0_nodeid = < sn0_nodeid>
cmn_hns_sam_control.hn_cfg_sn1_nodeid = < sn1_nodeid>
cmn_hns_sam_control.hn_cfg_sn2_nodeid = < sn2_nodeid>

##################
# OR
##################

##################
# Direct Mapping mode
##################
# Program each HNF to Direct Mapping mode
# With 32 HNFs target 4 SNs

for (hnf_num=0, hnf_num = hnf_num +1, hnf_num <32) begin        # begin
 # Group 8 HNFs and assign 1 SN target for all HNFs in that group
cmn_hns_sam_control.hn_cfg_sn0_nodeid = <sn#{hnf_num%8}_nodeid>
  end
```

4.4.3.2.2 RNSAM programming for Prefetch Target to a non-hash region

The following is an example for programming the RN SAM for a Prefetch Target to a non-hash region.

Example 4-44: RN SAM programming for a Prefetch Target

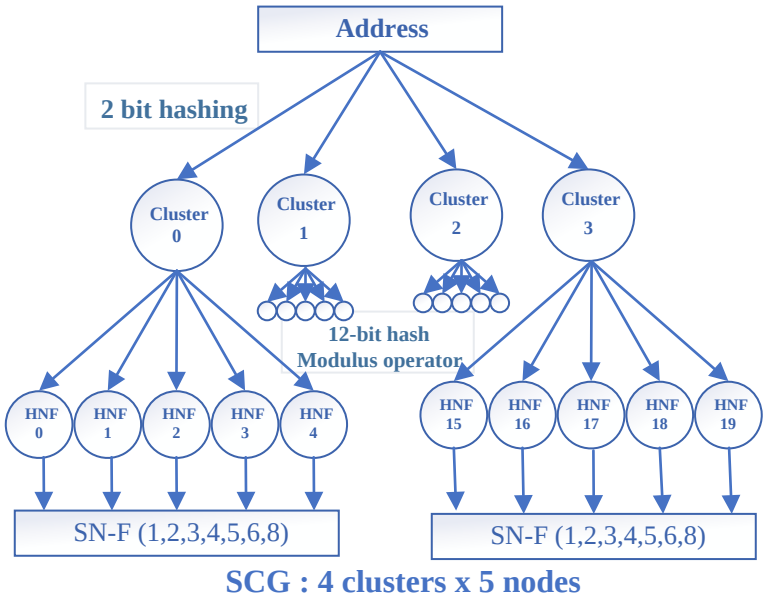
```
# Non-hashed region programming for SCG0 (2-129)
sam_scg0_prefetch_nonhashed_mem_region_cfg1_reg0.
 scg0_prefetch_nonhash_reg0_region_valid = 1'b1;
sam_scg0_prefetch_nonhashed_mem_region_cfg1_reg0.
 scg0_prefetch_nonhash_reg0_base_address = <start address of the region>;
```

```
sam_scg0_prefetch_nonhashed_mem_region_cfg2_reg0.
 scg0_prefetch_nonhash_reg0_end_address  = <end address of the region>;
sam_scg0_prefetch_nonhashed_mem_region_cfg2_reg0. scg0_prefetch_nonhash_reg0_tgtid =
 <SN nodeID>
```

4.4.3.2.3 CMN-700 SAM programming for Hierarchical Hashing

This is an example of Hierarchical hashing of a CMN-700 system configured for 4 clusters, 32HNFs with CALs, and HNSAM in 3-SN mode.

Figure 4-668: Hierarchical hashing configuration



- First hierarchy of hashing (clusters) is only power of two (supported clusters #2,4,8,16,32).
- Second hierarchy of hashing (nodes) is non-power of two (supported nodes # [2-32]).

First-level Hierarchy hashing:

Hashing for four clusters:

- Number of Address bits in select: 2
- select[0] = (6^8^10^...^50)
- select [1] = (7^9^11^...^51)

Second-level Hierarchy hashing

12 bit hashing:

- hash12[11:0] = {2'b00,Addr'[51:42]} ^ Addr'[41:30] ^ Addr'[29:18] ^ Addr'[17:6]
- select[6:0] = ({ hash12[11] ^ hash12[0],

  hash12[10] ^ hash12[1],

hash12[9] ^ hash12[2],

hash12[8] ^ hash12[3],

hash12[7] ^ hash12[4],

hash12[6] ^ hash12[5],

hash12[5:0] } * num_hnf) >> 12

Example 4-45: RNSAM programming with CAL

For each RNSAM:

```
#Hierarchical hashing config:
sys_cache_grp_region0.region0_base_addr = <base address for SCG>
                                                        # Set HTG0 Base Address
sys_cache_grp_region0.region0_valid = 1'b1;
                                                        # Set HTG0 Valid
hashed_tgt_grp_cfg2_region0.region0_end_addr = <end address for SCG>
                                                        # Set HTG0 End Address
hashed_target_grp_hash_cntl_reg0.htg_region0_hierarchical_hash_en = 1'b1;
                                                        # Set Hier hashing for HTG0
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_hash_clusters = 6'h4;
                                                        # 4 clusters in HTG0
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_hash_nodes = 6'h10;

                                                        # 16 nodes in a cluster
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_enable_address_striping = 3'b010;

                                                        #strip bit[7:6] for 4
 clusters
sys_cache_group_hn_count.scg0_num_hnf = 8'h40 ;
                                                        # 64 HNs in HTG0
sys_cache_grp_cal_mode_reg0.scg0_hnf_cal_mode_en = 1'b1
                                                        # enable CAL mode

##################
# With CAL2 program the 64 HNF targetIDs - 4 clusters - 16 targetIDs per cluster
# sys_cache_grp_hn_nodeid_reg[3:0]     covers HNF targetIDs 15:0 for cluster0
# sys_cache_grp_hn_nodeid_reg[7:4]     covers HNF targetIDs 31:16 for cluster1
# sys_cache_grp_hn_nodeid_reg[11:8]    covers HNF targetIDs 47:32 for cluster2
# sys_cache_grp_hn_nodeid_reg[15:12]   covers HNF targetIDs 63:48 for cluster3
##################
for (clustr=0, clustr = clustr+1, clustr<4) begin # 4 clusters with 16 HNF nodeIDs
 each
 for (indx=0, indx=indx+1, indx <4) begin   # 4 regs for each cluster for 16 HNF
 nodeIDs
   for (id=0, id=id+1, id <4) begin         # each reg covers 4 nodeIDs
         # 64 HNF targetIDs (cluster0,1,2,3)
     sys_cache_grp_hn_nodeid_reg#{4*clustr+indx}.nodeid_#{16*clustr+4*indx+id}
         = <HNF#{16*clustr+4*indx+id} {targetID[10:1],1'b0} >
         # upper 64 HNF targetIDs(not needed in CAL mode)
      end
   end
end

##################
# Prefetch SN targets in CAL mode
##################
sys_cache_grp_sn_attr.sn_hash_addr_bits_sel_sys_cache_grp0 = 3'b000;
sys_cache_grp_sn_attr.sn_mode_sys_cache_grp0 = 3'b001;
sys_cache_grp_sn_sam_cfg0.scg0_top_address_bit0 = 6'd33;
            # depends on customer memory map
```

```
sys_cache_grp_sn_sam_cfg0.scg0_top_address_bit1 = 6'd34;
             # depends on customer memory map
-----------------------------

# Total of 32 sys_cache_grp_sn_nodeid_regs available to program the SN targetIDs
# 1 sys_cache_grp_sn_nodeid_regs covers 4 SN targetIDs
# 4 sys_cache_grp_sn_nodeid_regs covers up to 16 SN targetIDs in a cluster
# (but don't need all regs)
# In 3-SN mode, just need reg0 to program 3-SN targetIDs (reg1-3 not needed) in
 cluster0.
# reg4, 8 and 12 cover HNs for cluster 1,2,3 respectively
# Cluster0 uses 4 HN regs to cover 16 HN TargetIDs, similarly 4
 sys_cache_grp_sn_nodeid_regs
# are allocated for cluster0 .Since we have only 3 SNs in each cluster we program
 only
# sys_cache_grp_sn_nodeid_reg0
-----------------------------
for (indx=0, indx=indx+4, indx<16) begin        # SN mapping for the 4 clusters
  sys_cache_grp_sn_nodeid_reg#{indx}.sn_nodeid_0 = <clstr#{indx/4}_sn0>
   sys_cache_grp_sn_nodeid_reg#{indx}.sn_nodeid_1 = <clstr#{indx/4}_sn1>
   sys_cache_grp_sn_nodeid_reg#{indx}.sn_nodeid_2 = <clstr#{indx/4}_sn2>
end
```

Program each HN-F in a cluster to 3-SN mode:

```
cmn_hns_sam_control.hn_cfg_three_sn_en = 1'b1;
cmn_hns_sam_6sn_nodeid.hn_hash_addr_bits_sel = 3'b000;
cmn_hns_sam_control.hn_cfg_sam_top_address_bit0 = 6'd33;
             # depends on customer memory map
cmn_hns_sam_control.hn_cfg_sam_top_address_bit1 = 6'd34;
             # depends on customer memory map

cmn_hns_sam_control.hn_cfg_sn0_nodeid = <clstr_sn0_nodeid>
cmn_hns_sam_control.hn_cfg_sn1_nodeid = <clstr_sn1_nodeid>
cmn_hns_sam_control.hn_cfg_sn2_nodeid = <clstr_sn2_nodeid>
```

The following programming is needed because of number of cluster(4) and hierarchical hashing:

```
# shift SLC/SF set address bits
cmn_hns_pa2setaddr_sf.setaddr_startbit_sf = 4'b1000;
                # 4 clusters: Setaddr starts from PA[8]
cmn_hns_pa2setaddr_slc.setaddr_startbit_sf = 4'b1000;
                                              # 4 clusters: Setaddr starts from
 PA[8]
```

For each MXP connected to the HNF:

```
# shift SLC/SF set address bits
por_mxp_p#{index}_pa2setaddr_slc.setaddr_startbit_slc_p#{index} = 4'b1000;
                # 4 clusters: Setaddr starts from PA[8]
por_mxp_p#{index}_pa2setaddr_sf.setaddr_startbit_sf_p#{index} = 4'b1000;
                # 4 clusters: Setaddr starts from PA[8]
```

4.4.3.2.4 Programming sequence to enable CXL.mem regions inside HNSAM

To define CXL memory regions, we must enable the following HNSAM user parameters:

```
# POR_HNSAM_NUM_HTG_PARAM = 2 (two regions)
# Region0 (HTG0) for 2 CCGs, CCG0, CCG1
```

```
# For each HNF:
cmn_hns_sam_htg_cfg1_memregion0.htg_region_valid0 = 1'b1;
cmn_hns_sam_htg_cfg1_memregion0.htg_region_base_addr0 = <start address of the
 region>;
cmn_hns_sam_htg_cfg2_memregion0.htg_region_end_addr0 = <end address of the region>;
cmn_hns_sam_htg_cfg3_memregion0.htg0_sn_mode = 3'b111;
                                                (Aggregated CCG SA selection
 function)
cmn_hns_sam_htg_cfg3_memregion0.htg0_sa_device_interleave_cntl = 4'b0000;
                                                                 (64 byte
 interleaved)
cmn_hns_sam_htg_cfg3_memregion0.htg0_sa_ports_cnt = 3'b001       (2 CXL links)
cmn_hns_sam_ccg_sa_nodeid_reg0.cxg_sa_nodeid_0 = <CCG nodeID0>
cmn_hns_sam_ccg_sa_nodeid_reg0.cxg_sa_nodeid_1 = <CCG nodeID1>
```

For one CXL link, we can either use HNSAM HTG or HNSAM non-hashed region:

```
# Region1 Hashed HTG1 for CCG2 - HTG programming
# For each HNF:
cmn_hns_sam_htg_cfg1_memregion1.htg_region_valid1 = 1'b1;
cmn_hns_sam_htg_cfg1_memregion1.htg_region_base_addr1 = <start address of the
 region>;
cmn_hns_sam_htg_cfg2_memregion1.htg_region_end_addr1 = <end address of the region>;
cmn_hns_sam_htg_cfg3_memregion1.htg1_sn_mode = 3'b111;
                                                 (Aggregated CCG SA selection
 function)
cmn_hns_sam_htg_cfg3_memregion1.htg1_sa_device_interleave_cntl = 4'b0000;
                                                                 (64 byte
 interleaved)
cmn_hns_sam_htg_cfg3_memregion1.htg1_sa_ports_cnt = 3'b000       (1 CXL links)
cmn_hns_sam_ccg_sa_nodeid_reg0.cxg_sa_nodeid_2 = <CCG nodeID>

# OR

# Region1 - non-hashed for CCG2 - Non-hashed region programming
# Foreach HNF:
cmn_hns_sam_memregion0.valid = 1'b1;
cmn_hns_sam_memregion0.base_addr = <start address of the region>;
cmn_hns_sam_memregion0_end_addr.end_addr = <end address of the region>;
cmn_hns_sam_memregion0.range0_nodeid = <CCG nodeID>
```

4.4.3.2.5 Multichip flat, 64B interleaving across 4 chips: (tgt_type = HN-Fs)

32 HNF's per chip, 128HNF across 4 chips, 2 CCGs connecting any two chips

Example 4-47: RNSAM programming (without CAL):

For each RNSAM:

```
sys_cache_grp_region0. region0_base_addr = <base address for SCG>
                                                  # Set HTG0 Base Address
sys_cache_grp_region0. region0_valid = 1'b1;
                                                   # Set HTG0 Valid
hashed_tgt_grp_cfg2_region0. region0_end_addr = <end address for SCG>
                                                   # Set HTG0 End Address
sys_cache_group_hn_count. scg0_num_hnf = 8'h80 ;
                                                   # 128 HNs in HTG0
sys_cache_grp_cal_mode_reg0.scg0_hnf_cal_mode_en = 1'b0
                                                   # do not enable CAL mode

##################
# Program the 128 HNF targetIDs
• HNF node id's on the local chip are programmed.
```

```
• HNF node id's on the remote chips are don't care (as these are overridden by CCG
 nodes).
• Set cpa_en = 1'b1 for the remote HNF's and program the cpag id for the remote
 HNF's.
##################
for (indx=0, indx=indx+1, indx <16) begin              # 32 registers for all the
 HNF's
 for (id=0, id=id+1, id <4) begin                     # 4 HNF's per each register
   sys_cache_grp_hn_nodeid_reg#{indx}. nodeid_#{4*indx + id} = <HNF#{4*indx + id}>
    hashed_target_grp_hnf_nodeid_reg#{16 + indx}.nodeid_#{64 + 4*indx + id}
    = <HNF#{64 + 4*indx + id}>
  end
end

sys_cache_grp_hn_cpa_en_reg.hash_cpa_en = 64'hEEEEEEEEEEEEEEEE
                                                    (cpa enable for 64 HNF's (0 - 63))
hashed_target_grp_hnf_cpa_en_reg1. htg_hnf_cpa_en1 = 64'hEEEEEEEEEEEEEEEE
                                                    (cpa enable for 64 HNF's (64 -
 127))
sys_cache_grp_hn_cpa_grp_reg.enable_multi_cpa_grp_scg0 = 1'b1

for (indx=0, indx=indx+1, indx <16) begin             # 16 registers (16 x 8 = 128)
  for (id=0, id=id+1, id <8) begin                   # 8 CPAG ids for each register
    hashed_target_grp_cpag_perhnf_reg{index}.htg_cpag_hnf{index*8 + id} = <CPAG for
 HNF{index*8 + id}>
  end
end

## program the CPAG's

cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_grp1_add_mask = <addr mask for the CPAG1>
cml_port_aggr_grp2_add_mask = <addr mask for the CPAG2>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.cpag_valid1 = 1'b1
cml_port_aggr_ctrl_reg.cpag_valid2 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_ctrl_reg.num_cxg_pag1 = 3'b001 (2 CCG ports)
cml_port_aggr_ctrl_reg.num_cxg_pag2 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0. pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0. pag_tgtid1 = <CCG-1>
cml_port_aggr_grp_reg0. pag_tgtid2 = <CCG-2>
cml_port_aggr_grp_reg0. pag_tgtid3 = <CCG-3>
cml_port_aggr_grp_reg0. pag_tgtid4 = <CCG-4>
cml_port_aggr_grp_reg1. pag_tgtid4 = <CCG-5>
```

4.4.3.2.6 PCIe IO Traffic examples (RNI to HNF)

The following table describes the PCIe IO traffic examples for local and remote interleaving.

Table 4-684: PCIe IO traffic interleaving examples

Traffic Examples	Local vs remote	Interleave granularity	RN SAM config
4.4.3.2.6.1 HNF interleave across local/remote HNF at cache line interleaving of 64bytes on page 1130	local/remote flat	<4K	HTG (addr based) CPAG (axid)
4.4.3.2.6.2 HNF interleave across local/remote HNF: 4K interleaving across chips and 64 bytes interleaving across HNFs on page 1130	local/remote flat	>=4K	HTG (addr based) CPAG (axid)

Traffic Examples	Local vs remote	Interleave granularity	RN SAM config
4.4.3.2.6.3 Remote NUMA (non-hash region to CPAG) on page 1131	remote NUMA	NA	non-hash CPAG (axid/addr)

4.4.3.2.6.1 HNF interleave across local/remote HNF at cache line interleaving of 64bytes

The following example is of two chips with 16 HN-Fs on each, using simple power of two hashing.

Example 4-48: Enabling interleave

```
##################
# Set up SCG0
##################
sys_cache_grp_region0. region0_valid       = 1'b1;
                         # Set SCG0 Valid
sys_cache_grp_region0. region0_target_type = 3'b000;
                                             # target type HNF
sys_cache_grp_region0. region0_base_addr   = <base address for SCG>;
                                             # SCG0 Base Address
sys_cache_grp_region0. region0_size        = <size of the SCG>;
                                             # SCG0 size
sys_cache_group_hn_count. scg0_num_hnf     = 8'h20;
                                             # 32 HNs in SCG0

##################
# Program the 32 HNF targetIDs -
# sys_cache_grp_hn_nodeid_reg[7:0]   covers 32 HNF targetIDs for cluster0
##################
for (indx=0, indx=indx+1, indx <8) begin          # 8 regs for 32 HNF nodeIDs
 for (id=0, id=id+1, id <4) begin                 # each reg covers 4 nodeIDs
   sys_cache_grp_hn_nodeid_reg#{indx}. nodeid_#{4*indx + id} = <HNF#{4*indx + id}
 targetID >
  end
end

sys_cache_grp_hn_cpa_en_reg.hash_cpa_en = 32'hCCCCCCCC
                                          (cpa enable for 32 HNF's (0 - 31)
sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg0 = <CCG0>

## program the CPAG's
cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0. pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0. pag_tgtid1 = <CCG-1>
```

4.4.3.2.6.2 HNF interleave across local/remote HNF: 4K interleaving across chips and 64 bytes interleaving across HNFs

The following example is of two chips with 16 HN-Fs on each. To enable 4K interleaving across local and remote chips, you must enable Hierarchical hashing and Cluster mask.

Example 4-49: Hierarchical hashing config

```
sys_cache_grp_region0. region0_base_addr = <base address for SCG>
                                             # Set HTG0 Base Address
sys_cache_grp_region0. region0_valid = 1'b1;
```

```
                                                        # Set HTG0 Valid
hashed_tgt_grp_cfg2_region0. region0_end_addr = <end address for SCG>
                                                        # Set HTG0 End Address
hashed_target_grp_hash_cntl_reg0. htg_region0_hierarchical_hash_en = 1'b1;
                                                        # Set Hier hashing for HTG0
hashed_target_grp_hash_cntl_reg0. htg_region0_hier_hash_clusters = 6'h2;
                                                      # 2 clusters in HTG0 (2 chips)
hashed_target_grp_hash_cntl_reg0. htg_region0_hier_hash_nodes = 6'h10;
                                                        # 16 nodes in a cluster
hashed_target_grp_hash_cntl_reg0. htg_region0_hier_enable_address_striping = 3'b001;

                                                        #strip bit[6] for 2 clusters
hashed_target_grp_hash_cntl_reg0. hier_cluster_mask = 4'b0110;
                                                        # 4K interleaving
sys_cache_group_hn_count. scg0_num_hnf = 8'h20 ;
                                                        # 32 HNs in HTG0

for (clustr=0, clustr = clustr+1, clustr<2) begin  # 2 clusters with 16 HNF nodeIDs
 each
 for (indx=0, indx=indx+1, indx <4) begin         # 4 regs for each cluster for 16
 HNF nodeIDs
   for (id=0, id=id+1, id <4) begin               # each reg covers 4 nodeIDs
         # 32 HNF targetIDs (cluster0,1)
     sys_cache_grp_hn_nodeid_reg#{4*clustr+indx}. nodeid_#{16*clustr+4*indx+id }
   = <HNF#{16*clustr+4*indx+id} targetID >
   end
  end
end

sys_cache_grp_hn_cpa_en_reg.hash_cpa_en = 32'hFFFF0000
                                                      (cpa enable for 32 HNF's (0 -
 31))
sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg0 = <CCG0>

## program the CPAG's
cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0. pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0. pag_tgtid1 = <CCG-1>
cml_port_aggr_grp_reg0.cpag_axid_hash_en = 1'b1
```

4.4.3.2.6.3 Remote NUMA (non-hash region to CPAG)

The following example is of two chips with 16 HN-Fs on each, where the HN-Fs on the remote chip are configured as NUMA through the non-hashed region and CPAG (axid based hashing).

Example 4-50: Configuring non-hash region

```
non_hash_mem_region_reg0.base_addr = <base addr>
non_hash_mem_region_reg0.valid = 1'b1
non_hash_mem_region_cfg2_reg0.end_addr= <end addr>
cml_port_aggr_mode_ctrl_reg.cpag_en = 1'b1
cml_port_aggr_mode_ctrl_reg.cpag_grpid = CPAG0

# program the CPAG's
cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0. pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0. pag_tgtid1 = <CCG-1>
```

```
cml_port_aggr_grp_reg0.cpag_axid_hash_en = 1'b1
```

4.4.4 Program the dual DAT and RSP channel selection scheme

Boot-programmable registers control the dual DAT and RSP channel selection scheme by forming a TargetID LUT. Programming these registers at boot overrides the default channel selection scheme.

About this task

When the `EN_2X_[DAT/RSP]_VC` parameters are enabled, a default channel selection scheme is applied to the respective CHI channels. For more information about the scheme, see 3.1.4.2 Dual CHI channel selection on page 69. This scheme is active until you configure the dual channel registers.

A constraint applies to the channel assignment for the RN-F or RN-I and HN-D that is used to program the dual channel registers. You must program these nodes when the`EN_2X_[DAT/RSP]_VC` parameters use the same channel as the default channel selection scheme, according to the MXP XID. For example, if the HN-D node is connected to $XP_{(1,0)}$, it has an odd XID (XID = 1), so by default targets channel 1. When programming is complete, the HN-D node ID must have CHN_SEL set to channel 1 in the LUT.

Procedure

1. Identify and classify the targets that must map to channel 0 and channel 1
2. Program the TargetID and CHN_SEL fields in por_mxp_multi_mesh_chn_sel_* registers with the TargetIDs to map to the required channel, 0 or 1.
3. Set the VALID bit in each por_mxp_multi_dat_mesh_chn_sel_*register to indicate that you have configured valid TargetIDs in each register.
4. Set the multi_mesh_chn_sel_programmed bit in the por_mxp_multi_mesh_chn_ctrl register to indicate that channel selection configuration is complete.
5. Repeat steps 1-4 for each MXP in the mesh and ensure that the same values are programmed in the por_mxp_multi_mesh_chn_sel_* registers for all MXPs.

4.4.5 Program non-XY routing registers

To configure the behavior of the CMN-700 non-XY routing feature at boot, you must program the por_mxp_xy_override_sel_* registers for each MXP.

Before you begin

You must ensure that your mesh configuration is free of deadlocks when using the non-XY routing feature. For more information about how to avoid deadlocks when setting up this feature, see 3.4.12.2 Rules for avoiding deadlocks in non-XY routing on page 180.

About this task

When you enable this feature by setting the configuration parameter, the default XY route is applied. The default scheme is active until it is reconfigured by programming the por_mxp_xy_override_sel_* registers.

The following constraints apply to this process:

- A maximum of 16 source-target pairs are supported.
- You must configure all MXPs in your mesh with the same set of <SRCID> and <TGTID> values. These values correspond to the set of source-target pairs that is overridden, so must be identical in all MXPs. Additionally, the same por_mxp_xp_override_sel_* register index (0-7) and the same register fields must be used for the same SrcID-TgtID pairs in every XP.
- You must only set the XY override and YX turn bits in the registers for the overridden MXP or MXPs.
- All devices that are attached to a CAL have the same routing scheme applied. Therefore all devices that are attached to a CAL are either part of the non-XY scheme or not. Selection of partial devices behind a CAL for non-XY routing is not allowed.

Procedure

1. Identify the node source-target pairs and the MXP or MXPs to be overridden.
2. Program the <SRCID> and <TGTID> bits in the relevant por_mxp_xy_override_sel_* registers with the IDs of the chosen source-target pairs.
3. Set the VALID bit in each por_mxp_xy_override_sel_* register to indicate that valid source-target pairs are configured in each register.
4. Repeat steps 1-3 for each MXP in the mesh, ensuring that you program the same values for all MXPs.
5. Set the XY_OVERRIDE_ENABLE bit or bits in any MXPs where you want to override the XY route for a given source-target pair.
6. Set the YX_TURN_ENABLE bit or bits in any MXPs where the YX turn is allowed for a given source-target pair.

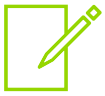
4.4.6 RN-I and HN-I PCIe programming sequence

To ensure proper PCIe functionality, software must complete the following programming before any non-configuration access to the RN-I or HN-I.

About this task

When setting up PCIe RN-Is and HN-Is, the entire PCIe configuration space of an RC must be mapped to a single HN-I address region. HN-I has a SAM that can be configured for up to three address regions. All address regions that are not configured into these three regions are considered to be the default address region. For more information about configuring the HN-I SAM, including example configurations, see 3.4.8 HN-I SAM on page 163.
If you map an HN-I SAM address region to a PCIe subordinate, you must map all address regions of that HN-I SAM to PCIe subordinates.

Throughout the following procedure, address region X or address region Y can refer to any of the four address regions (0, 1, 2, or 3).


Note

- Peer-to-peer writes to HN-P always assume that traffic is directed to the PCIe endpoint memory space. Therefore, HN-P does not give early write completions for any write request.
- Peer-to-peer reads to HN-P also assume traffic is directed to PCIe endpoint memory space.

Procedure

1. If there is a PCIe-RC attached to the RN-I then set the pcie_mstr_present field of the por_rni_cfg_ctl register. This programming indicates that one or more PCIe managers are present upstream. Also, dis_awid_to_hni_hnp_cxra bit in the por_rn{i,d}_cfg_ctl register, must remain set to 0 to enable passing the AxID signal to HN-I and HN-P.
2. Program the por_hni_sam_addrregion{0,1,2,3}_cfg registers so that the PCIe configuration space falls under one of the four HN-I address regions. This programmed address region is referred to as address region X.
3. Set only one of the following bits in por_hni_sam_addrregionX_cfg for address region X:

   ser_devne_wr

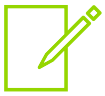
   Set this bit if the PCIe configuration space is marked as the Arm Device-nGnRnE memory type. If this bit is set, HN-I serializes all Device-nGnRnE writes to address region X. The HN-I does not send any other write requests with the same AWID as an outstanding Device-nGnRnE write.

   ser_all_wr

   Set this bit if the PCIe configuration space cannot be marked as Arm memory type of Device-nGnRnE. If this bit is set, HN-I serializes all writes targeting Address Region X.
4. Clear the pos_early_wr_comp_en bit of the por_hni_sam_addrregionY_cfg register for address region Y.
   Address region Y is the region in which PCIe EP memory space (posted traffic) is programmed. If the pos_early_wr_comp_en bit is cleared, HN-I does not give early write completions for any write requests targeting address region Y.
   For address region Y, in which EP memory space is programmed, the physical_mem_en field of the por_hni_sam_addrregionY_cfg register can be set to enable Normal memory behavior of posted traffic.

4.4.7 CML programming

The system must be programmed to enable correct operation with CML.


Note

Programming steps specified in this section are applicable to the CCG unit.

4.4.7.1 CML-related programmable registers

This section contains a list of CML programmable registers.

RA

- RA SAM address region registers (por_ccg_ra_sam_addr_region_reg<X>)
- LDID to RAID LUT registers:
  - por_ccg_ra_rnf_ldid_to_ovrd_ldid_reg<X>
  - por_ccg_ra_ha_ldid_to_exp_raid_reg<X>
  - por_ccg_ra_rnf_ldid_to_exp_raid_reg<X>
  - por_ccg_ra_rni_ldid_to_exp_raid_reg<X>
  - por_ccg_ra_rnd_ldid_to_exp_raid_reg<X>
- RAID or HAID to LinkID LUT registers:
  - por_ccg_ra_agentid_to_linkid_reg<X>
  - por_ccg_ra_agentid_to_linkid_val
- Auxiliary Control register (por_ccg_ra_aux_ctl)
- Configuration Control register (por_ccg_ra_cfg_ctl)
- PortID assignment (por_ccg_ra_node_info)

HA

- HAID register (por_ccg_ha_id)
- RAID to LDID LUT registers
  - por_ccg_ha_rnf_exp_raid_to_ldid_reg<X>
- RAID or HAID to LinkID LUT registers:
  - por_ccg_ha_agentid_to_linkid_reg<X>
  - por_ccg_ha_agentid_to_linkid_val
- Auxiliary Control register (por_cxg_ha_aux_ctl)
- RN SAM

HN-F

- LDID to CHI NodeID registers (cmn_hns_rn_cluster<X>_physid_reg<Y>)

RN SAM

- RN SAM
- CML Port Aggregation Mode Enable and Control registers:
  - cml_port_aggr_grp<X>_reg
  - cml_port_aggr_mode_ctrl_reg

- CML Port Aggregation Mask register (cml_port_aggr_grp<X>_add_mask)


Note

Additional registers are used to enable PCIe write tunneling and streaming across CML_SMP link.

4.4.7.2 Bring up a CML system

Use the following sequence to bring up a CML system.

Procedure

1. Discover and bring up the local CMN-700 system.
   See 4.4.7.2.1 Discover and bring up a local CMN-700 system on page 1136
2. CML_SMP Connection:
   CML_SMP connection can be enabled by connecting multiple CMN-700s through a CML_SMP link. Follow CMN-700 discovery mechanisms to discover node types and capabilities present on each CMN-700 . Go to 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.
3. CXL Connection
   If CML link is configured to attach a CXL device, go to 4.4.7.8 Program CML system to enable CXL communication on page 1151.

Next steps

For information about the programming requirements that are necessary for communication between CCIX components, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

4.4.7.2.1 Discover and bring up a local CMN-700 system

Use this process to bring up a local CMN-700 system during a full bring up of a CML system.

About this task

This procedure is the first step in the sequence to bring up a CMN-700 CML system. For the full sequence, see 4.4.7.2 Bring up a CML system on page 1136.

Procedure

1. Complete the CMN-700 discovery mechanism to discover node types, their corresponding locations (node IDs), and their logical IDs.
   3.4.1 Node ID mapping on page 101 and 3.5 Discovery on page 193 define the discovery mechanism. Node types that are relevant to CML-specific programming are RN-Fs, RN-Is, RN-Ds, HN-Fs, and CML gateway blocks (RA, HA, and LA).
2. Bring up the local system to allow normal local operations. To bring up all local non-CML components (HN-F, HN-I, HN-D, HN-P, RN-I, SN-F, and XP):
   a) Complete CMN-700 boot time programming.

See 4.4.1 Boot-time programming sequence on page 1120.

b) Program RN SAM with the local address map.
See 4.4.3.1 Program the SAM on page 1121.

4.4.7.3 Program CML system to enable CML_SMP communication

Use this procedure to enable CML communication between different multichip entities. The steps can be completed in any order.

Before you begin

Before enabling CML communication, you must first complete the bring up process. See 4.4.7.2 Bring up a CML system on page 1136
Program CML link for CML_SMP connection

About this task

The terms *link*, *CML link*, and *CML protocol link* that are used in subsequent sections refer to a logical link. Only a single logical link, link 0, is supported with CMN-700.

Procedure

- Program the auxiliary control and configuration control registers.
  For more information about specific optional functionality, see 4.4.7.3.1 Options when programming CCG auxiliary control and configuration control registers on page 1137.
- Program IDs for local RAs and HAs.
  See 4.4.7.3.2 Assign IDs for local RAs and HAs on page 1138.
- Program remote agents:
  - Assign LinkIDs to remote CML protocol links. For more information, see 4.4.7.3.3 Assign LinkIDs to remote CML protocol links on page 1139.
  - Assign LDIDs to remote caching agents. For more information, see 4.4.7.3.4 Assign LDIDs to remote caching agents on page 1140.
  - Program RA SAMs. See 4.4.7.3.5 Program RA SAM on page 1140.
  - Program RNSAM in HAs. See 4.4.7.3.6 Program RN SAM in HA on page 1141.
  - Program CML protocol link control registers. For more information, see 4.4.7.3.8 Program CML protocol link control registers on page 1141.
  - Program CPA functionality in RN SAM, if using CPAGs. For more information, see 4.4.7.3.10 Program CPA functionality in RN SAM on page 1142.
  - Program CPA functionality in HN-F SAM, if using CPAGs. See 4.4.7.3.11 Program CPA functionality in HN-F SAM on page 1143.

4.4.7.3.1 Options when programming CCG auxiliary control and configuration control registers

CMN-700 has optional CML functionality that can be enabled when you program the CCG auxiliary control and configuration control registers. There are specific constraints that you must follow when enabling this functionality for your CML system.

CMN-700 supports the following functionality in the CCG auxiliary and configuration control registers:

- SMP mode
- CXSA mode

SMP mode

SMP connection requires CCG block and is enabled by setting the lnk<X>_smp_mode_en bit in the following registers:

- por_ccg_ra_cxprtcl_link<X>_ctl
- por_ccg_ha_cxprtcl_link<X>_ctl

The SMP mode programming must be the same in RA and HA for a specific CML protocol link. Also, all CCG pairs that can communicate with each other must be configured in the same way.

4.4.7.3.2 Assign IDs for local RAs and HAs

Use this procedure to assign identifiers (IDs) for local RAs and HAs and configure them in the relevant registers.

About this task

This task is part of the requirements to enable communication between local and remote CCIX agents. For the full list of requirements, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

Procedure

- Assign *Requesting Agent IDs* (RAIDs) for local RAs.
  a) Program all the local RAIDs in the following registers for all RAs and set the corresponding valid bit:
     - por_ccg_ra_rnf_ldid_to_exp_raid_reg
     - por_ccg_ra_rni_ldid_to_exp_raid_reg
     - por_ccg_ra_rnd_ldid_to_exp_raid_reg
     - por_ccg_ra_ha_ldid_to_exp_raid_reg
     - por_ccg_ra_hns_ldid_to_exp_raid_reg

     This programming sets up the LDID to RAID LUT.


Note

- HA's LDID to RAID entry must be programmed only for CCG-HA connected to a CXL Type1 Device
- All HA's, irrespective of usage i.e. SMP or CXL, are part of the same logical ID space (LDID). So, RAID must be programmed at LDID of CCG-HA connected to CXL Type1 Device

- Program *Home Agent IDs* (HAIDs) for all local HAs into the por_ccg_ha_id registers that are present in each HA.
  Because all HAs can communicate with all local HNs (HN-F, HN-I, HN-D, and HN-P), they can have the same HAID.

  An HA and RA with the same ID must reside behind the same CML protocol link.

4.4.7.3.3 Assign LinkIDs to remote CML protocol links

Use this procedure to assign a unique LinkID to each remote CML protocol link with which a CML unit can communicate.

About this task


Note

CCG supports only one link and all valid agents must be programmed with LinkID 0.

This task is part of the requirements to enable communication between local and remote agents. For the full list of requirements, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

Remote agents, RAs or HAs, are identified using their RAID or HAID. Each remote RA or HA that a CML gateway can communicate with must be behind only one link.

It is only necessary for LinkIDs to be unique within a gateway block. Each CML has a respective LinkID space. Each remote link has its own CML protocol link control and status registers.

Procedure

1. Determine the LinkID of each remote agent, in other words the targets, of the CML.
2. Program these LinkIDs in the following registers, which are present in the RA and HA:
   - por_ccg_ra_agentid_to_linkid_reg<X>
   - por_ccg_ha_agentid_to_linkid_reg<X>


Note

CCG-LA does not have AgentID to LinkID register

This step sets up the AgentID (RAID or HAID) to LinkID LUT.

3. Set the respective valid bits in the following registers:
   - por_ccg_ra_agentid_to_linkid_val
   - por_ccg_ha_agentid_to_linkid_val

4.4.7.3.4 Assign LDIDs to remote caching agents

Use this procedure to assign a unique LDID for each remote caching agent (RN-F) that can send requests to HNs (HN-F, HN-I, HN-D, and HN-P).

About this task

This task is part of the requirements to enable communication between local and remote agents. For the full list of requirements, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

HN-Fs use the LDID of remote caching agents for SF tracking. LDID assignment is not required for non-caching RAs, for which snooping is not required.

Procedure

1. Program unique LDIDs for each remote caching agent in the por_ccg_ha_rnf_exp_raid_to_ldid_reg<X> register that is present in each HA. The LDID values for remote RN-Fs must be greater than those values that are used by the local RN-F nodes, unless you override local RN-F LDIDs by using the por_ccg_ra_rnf_ldid_to_ovrd_ldid_reg_0-127 register.
2. Set the ldid<X>_rnf bit, which marks the remote agent as a caching agent, and set the respective valid bit.
3. Program the HA NodeID at the LDID index of each remote RN-F in cmn_hns_rn_cluster<X>_physid_reg<Y> register in the HN-F. Set the appropriate attributes for remote bit, CPA enable, and CPA group information for each LDID.

4.4.7.3.5 Program RA SAM

Use this procedure to program the RA SAM, which generates the target ID for outbound requests.

About this task

This task is part of the requirements to enable communication between local and remote agents. For the full list of requirements, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

Procedure

Program the following properties for each remote HA into the por_ccg_ra_sam_addr_region_reg<X> register that is present in each RA, and set the corresponding valid bit.

- The base address of the address region and the corresponding size of the address region
- The HAID that requests for the address range are mapped to

4.4.7.3.6 Program RN SAM in HA

Use this procedure to program the RN SAM in each HA.

About this task

You can program HA RN SAM as part of local system bring up.

Procedure

Program the RN SAM present in each HA with the address and memory map of the local HNs.
For more information about RN SAM programming, see 4.4.3 RN SAM and HN-F SAM programming on page 1121.

4.4.7.3.7 Program RN SAM in CCG

Use this procedure to program the RN SAM in each CCG.

About this task

RN SAM is used to route the incoming request to respective Home Nodes. In hierarchical caching system, RN SAM is used to route the incoming snoops to respective Local Coherency Nodes (LCNs) as well.

For more information about RN SAM programming, see 4.4.3 RN SAM and HN-F SAM programming on page 1121.

4.4.7.3.8 Program CML protocol link control registers

Use this procedure to set up CML protocol links for a gateway block and configure the distribution of credits that the CML uses.

About this task

There is a protocol link control register for each CML protocol link that a given gateway block (RA, HA, and LA) can communicate with.

Procedure

Program the protocol link control (*prtcl_link) registers that are present in each RA and HA.
If CXSA mode is enabled, it is not necessary to program the protocol link control registers of the HA. In this case, the lnk0_num_snpcrds of the RA can be set to `4'hF`.

a) Set the lnk<X>_link_en bit for each protocol link that can be used in the future.

If this bit is not set, credits are not set aside for this link.

b) Program the lnk<X>_num_{snpcrds, reqcrds, datcrds} fields with the percentage of protocol credits that must be assigned or granted for a given link.
This step is optional. Default credits are equally assigned or granted to each enabled link as determined by the link enable bit (lnk<X>_link_en). For more information about credit distribution and the permitted configurations, see 4.4.7.3.9 CML protocol link credit distribution on page 1142.
You must ensure that the total percentage of credits that are allocated to all links does not exceed 100.

4.4.7.3.9 CML protocol link credit distribution

When setting up protocol links, you can specify the protocol link credit distribution. The distribution can be configured when programming the protocol link control registers.

The link enable bits (lnk<X>_link_en) of the RA and HA protocol link control (*prtcl_link<X>_ctl) registers determine how many links are active for a gateway block. By default, credits are equally assigned or granted to each enabled link. However, you can program these registers to configure the distribution of credits across multiple links.

The following table shows the number of links and allowed credit distribution percentages for that number of links.

Table 4-685: Number of links and allowed credit distribution percentage

Number of links	Allowed credit distribution
1	100%
2	50% : 50%
	25% : 75%
3	50% : 25% : 25%
	33% : 33% : 33%

After distributing credits based on the programmed percentage across all the links available, any remaining credits are allocated to link0. For example, link0 is allocated 44 credits while link1 and link2 are allocated 42 credits each for the 128 credit, 33% credit distribution configuration.

For more information about programming the protocol link control registers, see 4.4.7.3.8 Program CML protocol link control registers on page 1141.

4.4.7.3.10 Program CPA functionality in RN SAM

There is a specific sequence of programming steps that must be followed to set up the RN SAM to distribute requests to CPAGs.

About this task

This task is part of the requirements to enable communication between local and remote agents. For the full list of requirements, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

To enable CPA, you must program registers in both the RN SAM and HN-F SAM. For the programming sequence for the HN-F SAM registers, see 4.4.7.3.11 Program CPA functionality in HN-F SAM on page 1143.

There are certain rules and restrictions that apply to how CPA can be enabled in a CMN-700 system. See 3.4.6.8 SAM support for CML Port Aggregation on page 135.

In a two-chip system with CPA enabled, the RN SAM CPA mask of Chip 0 and HN-F SAM CPA mask of Chip 1 must be programmed to match. Similarly, the RN SAM CPA mask of Chip 1 must match the HN-F SAM CPA mask of Chip 0.

Procedure

1. Set the region<n>_pag_en field of the cml_port_aggr_mode_ctrl_reg register to 1 and region<n>_pag_grpid with the relevant group ID for each non-hashed memory region that belongs to a remote chip and is required to use CPA.
   This programming enables CPA mode for the non-hashed memory region and specifies the CPAG that requests must be hashed across.
2. Set each bit that must be used to hash requests across CML gateway to 1 in the addr_mask field of the cml_port_aggr_grp<n>_addr_mask register. Set each bit that must be masked from the hash function to 0.
   If CPA mode is enabled, RN SAM hashes PA bits[51:6] to distribute traffic between CML gateways. The PA is compared against the addr_mask field of the cml_port_aggr_grp<n>_addr_mask_register. Any PA bit that corresponds to a bit with a 0 value in the mask register is masked from hashing.
3. Program the characteristics of the CML gateways for each CPAG by completing the following steps:
   a) Program the number of CML gateways in each CPAG by setting the num_cxg_pag<n> field in the cml_port_aggr_ctrl_reg register.
   b) Program the CHI node ID of each CML gateway by setting the pag0_tgtid field of the same register.
      If two CML gateways are being used, then both node IDs must be programmed in this register.

   This programming is used by all regions that have CPA enabled.
4. Repeat the preceding steps for all RN SAMs in the chip.

4.4.7.3.11 Program CPA functionality in HN-F SAM

There is a specific sequence of programming steps that must be followed to set up the HN-F SAM to distribute snoop traffic to CPAGs.

About this task

This task is part of the requirements to enable communication between local and remote agents. For the full list of requirements, see 4.4.7.3 Program CML system to enable CML_SMP communication on page 1137.

To enable CPA, you must program registers in both the RN SAM and HN-F SAM. For the programming sequence for the RN SAM registers, see 4.4.7.3.10 Program CPA functionality in RN SAM on page 1142.

There are certain rules and restrictions that apply to how CPA can be enabled in a CMN-700 system. See 3.4.6.8 SAM support for CML Port Aggregation on page 135.

In a two-chip system with CPA enabled, the RN SAM CPA mask of Chip 0 and HN-F SAM CPA mask of Chip 1 must be programmed to match. Similarly, the RN SAM CPA mask of Chip 1 must match the HN-F SAM CPA mask of Chip 0.

Procedure

1. Set the following fields in the cmn_hns_rn_cluster<X>_physid_reg<Y> registers for each valid LDID in the system:

   nodeid_lid<X>_ra<Y>

   When using CPA, set this value to match the pag_tgtid0 field of the cmn_hns_cml_port_aggr_grp_reg<m> field of the corresponding CPAG.

   remote_lid<X>_ra<Y>

   Set this value to 1 if the RN-F is a remote requestor.

   cpa_grp_lid<X>_ra<Y>

   Set this value to the corresponding CPA group.

   srctype_lid<X>_ra<Y>

   Set this value to the appropriate CHI protocol version, either CHI-B, CHI-C, CHI-D or CHI-E.

   Local RN-F LDIDs must have the remote_ra<ldid> and cpa_en_ra<ldid> bits set to 0 and the corresponding CPA group ID set to 0.
2. Set each bit that must be used to hash snoop traffic across CML gateways to 1 in the addr_mask field of the cmn_hns_cml_port_aggr_grp<m>_addr_mask register. Set each bit that must be masked from the hash function to 0.
   If CPA mode is enabled, HN-F SAM hashes PA bits[51:6] to distribute snoop traffic between CML gateways. The PA is compared against the addr_mask field of the cml_port_aggr_grp<m>_addr_mask_register. Any PA bit that corresponds to a bit with a 0 value in the mask register is masked from hashing.
3. Program the characteristics of the CML gateways for each CPAG by completing the following steps:

a) Program the number of CML gateways in each CPAG by setting the num_cxg_pag<n> field in the cmn_hns_cml_port_aggr_ctrl_reg register.
b) Program the CHI node ID of each CML gateway into the pag_tgtid<n> fields in cmn_hns_cml_port_aggr_grp_reg0 and cmn_hns_cml_port_aggr_grp_reg1.

This programming is used by all regions that have CPA enabled.

4. Repeat the preceding steps for all HN-F SAMs in the chip.

4.4.7.3.12 CCLA to CCLA Direct connect mode

CCG enables you to directly connect the CXS interface from the CCLA on one CMN-700 to the CXS interface of the other. In this mode, you can connect these interfaces without going through the lower link layer and PHY controller IP. Use this mode in simulation environments for quick system bringup. Do not use this mode when external controller IP is present.

About this task

The following steps must be followed to enable this Direct connect mode.

1. Set ull_to_ull_en bit on both sides in CCLA por_ccla_ull_ctl register.
2. After ull_to_ull_en is set on both sides (step 1), set the send_vd_init bit on both sides in CCLA por_ccla_ull_ctl register.

The following steps must be followed to quiesce and bring down the link (at the end):

Procedure

1. Clear send_vd_init bit on both sides in CCLA por_ccla_ull_ctl register.
2. Clear ull_to_ull_en bit on both sides in CCLA por_ccla_ull_ctl register.
See 4.3.4.59 por_ccla_ull_ctl on page 427 and 4.3.4.60 por_ccla_ull_status on page 428.

4.4.7.3.13 Programming to enable tunneling of PCIe traffic

1. PCIe RN-I Programming:
   a. For every address region if the target_nide is CCG, program RN SAM non_hash_mem_region_reg#{index}. region#{index}_target_type register with PCI_CXRA target type
   b. Set the pcie_mstr_present field of the por_rni_cfg_ctl register. This programming indicates that one or more PCIe managers are present upstream.
   c. Address based port aggregation must be disabled for writes targeting a remote PCIe HN-P to preserve write ordering.
2. CCG RA programming:
   a. Set remote_rni_present field of the por_ccg_ra_aux_ctl register. This programming indicates that the HA on remote chip has RN-I instantiated inside it to support PCIe read burst and write gathering features.
3. CCG RN-I

a. Set the pcie_mstr_present field of the por_rni_cfg_ctl register. This programming indicates that one or more PCIe managers are present upstream.

b. Follow the programming sequence outlined in 4.4.6 RN-I and HN-I PCIe programming sequence on page 1133

The high bandwidth tunneling of PCIe traffic can be disabled for various reasons like

- RN-I is not present in remote CCG

The following programming must be followed to disable tunneling


Note

Any violation in below programming could result in violation with respect to PCIe ordering or forward progress guarantees.

1. PCIe RN-I/RN-D Programming
   a. For every address region if the target is CCG, program RN SAM non_hash_mem_region_reg#{index}. region#{index}_target_type register with RA target type
   b. Set the pcie_mstr_present field of the por_rni_cfg_ctl register. This programming indicates that one or more PCIe managers are present upstream.
2. CCG RA Programming
   a. Clear remote_rni_present bit in por_ccg_ra_aux_ctl register
   b. Set dis_rnid_tnl_retry_trk bit in por_ccg_ra_aux_ctl register
   c. Set dis_rnid_early_wrcomp bit in por_ccg_ra_aux_ctl register

4.4.7.3.14 Programming to enable PCIe write streaming through CML SMP link

1. PCIe RN-I Programming:
   - Set the pcie_mstr_present field of the por_rni_cfg_ctl register. This programming indicates that one or more PCIe managers are present upstream
   - The value of clstr_intlv_mask field of por_rni_aux_ctl register must be default value of 0x00 (4KB). Current CMN-700 version does not support any of the other interleaving granularities specified in this register. So, if the address interleaving between local and remote is less than 4K then this streaming mechanism will be use
2. RN SAM programming: If the address range targeted by PCIe traffic is striped between local and remote then the following programming is required.
   - Program region<X>_target_type field in RN SAM's sys_cache_grp_region<X> to HN-F ('b000)

   The following fields in RN SAM's hashed_target_grp_hash_cntl_reg<X> register must be programmed


Note

The following fields in RN SAM's hashed_target_grp_hash_cntl_reg<X> register must be programmed

- Program htg_region<X>_hier_cluster_mask with the interleaving granularity at which addresses are striped between local and remote memory. By default, this field is programmed to 0xF and programming any other value enables the streaming mechanism
- Program htg_region<X>_hier_hash_nodes with the number of HN-Fs per cluster
- Program htg_region<X>_hier_hash_clusters with the number of cluster
- Program htg_region<X>_hier_enable_address_striping based on the number of clusters
- Program htg_region<X>_hierarchical_hash_en to enable this mode


Note

For a case where an HTG address range is striped between local and remote, the entire chip could be considered as one cluster

- Program hashed_target_grp_hnf_target_type_override_cfg_reg<X> to override the HN-F target type with the CCG. RNI/D uses this information to enable tunneling vs streaming to remote chip.

3. HN-F and MXP programming:
   - Program HN-F and MXP with the same cluster information as RN SAM. For more information about HN-F and MXP programming, see 3.4.7.4 HN-F SLC and SF flexibile addressing on page 160 and 4.4.3.2 SAM programming examples on page 1123
4. CCG RA programming:
   - Clear disable_3hop_wr_flow field of the por_ccg_ra_aux_ctl register. By default, this bit is cleared.


Tip

For a case where interleaving is enabled, and interleaving granularity is 4K or greater then dis_hnf_logical_tgt_2hop bit in por_rni_aux_ctl register can be set to enable this streaming mechanism when targeting remote memory. This aux_ctl bit is applicable to entire RN-I and is not per HTG.

4.4.7.3.15 Program RA to HN-F snoop flow control

Program the num_outstanding_hns_snp field in por_ccg_ra_cfg_ctl register based on "min(HNS_NUM_ENTRIES_SNPQ_PARAM) / (number of RAs that can snoop HNS)", where

min(HNS_NUM_ENTRIES_SNPQ_PARAM) is the minimum depth of the incoming snoop queue at across all HN-F. .

About this task

"Number of RAs that can snoop HN-F" are all non-CXL RAs that can send snoops to each HN-F. Default value of this field is set to "HNS_NUM_ENTRIES_SNPQ_PARAM / (Number of all RAs in CMN)".

4.4.7.4 Program CMN-700 CML system at runtime

Use this procedure to program a CMN-700 CML system at runtime.

Procedure

1. Bring up a protocol link.
   See 4.4.7.5 Establish protocol link up between CML gateway and remote CML link on page 1148.
2. Add a protocol link in system coherency and DVM domains.
   See 4.4.7.7 Entry and exit protocol links from coherency domains and DVM domains on page 1150.


Note

If CXSA mode is enabled, this programming is not necessary.

3. Program the remote address range and corresponding RA node ID for each remote memory region in RN SAM present in CMN-700 RN-F, RN-I, and RN-D.
   RN SAM must not be programmed to target RA when enabled for CXSA mode.


Note

If the software can guarantee that there is no traffic to the remote address range until CML-related initial programming is complete and protocol links are up, then this programming should be done when programming RN SAMs with local address map.

4.4.7.5 Establish protocol link up between CML gateway and remote CML link

Use the following procedure to link up a CML gateway with a remote protocol link that the gateway can communicate with.

About this task

A protocol link can be established between each CML in CMN-700 and a corresponding remote CML link that the gateway can communicate with. The term *link* is used here to refer to a CML protocol link. Multiple protocol links can be set up simultaneously by extending this sequence for each link.

Procedure

1. Poll the lnk<X>_link_en bit in both RA and HA protocol link control (*prtcl_link<X>_ctl) registers to ensure that the link is enabled.
   If the link is enabled, then communication on the link can be established.
2. Ensure that the link is down and can accept a new link up request by polling the following bits:
   a) Poll the lnk<X>_link_up bits in the RA and HA protocol link control (*prtcl_link<X>_ctl) registers to ensure that they are clear
   b) Poll the lnk<X>_link_down bits in the RA and HA protocol link status (*prtcl_link<X>_status registers to ensure that they are set
   c) Poll the lnk<X>_link_ack bits in the RA and HA protocol link status (*prtcl_link<X>_status) registers to ensure that they are clear
3. Set the lnk<X>_link_req bits in the RA and HA protocol link control (*prtcl_link<X>_ctl) registers
   Setting this bit generates a request to link up, which in turn brings up the link.
4. Poll the following bits in the RA and HA protocol link status (*prtcl_link<X>_status) registers to ensure that the link up request is accepted:

   - lnk<X>_link_ack
   - lnk<X>_link_down

   The hardware acknowledges a link up request by setting the lnk<X>_link_ack bits and then clearing lnk<X>_link_down bits in RA and HA.

   Hardware acknowledgment of link up means that both sides are ready to receive and grant protocol credits.
5. Set the lnk<X>_link_up bit in the RA and HA protocol link control (*prtcl_link<X>_ctl) registers to instruct both sides to start granting credits.

Results

Link<X> is now up. Both sides can now exchange protocol credits and protocol messages.

4.4.7.6 Link down protocol link between CML gateway and remote CML link

Use this procedure to deactivate a protocol link between a CMN-700 CML gateway and a remote CML link.

Before you begin

Software must ensure that the following conditions are met before initiating a link down sequence:

- There are no outstanding transactions that require protocol message transfers across the link for their completion. This condition includes GIC-D in SMP mode.
- The protocol agents on both sides of the link are configured to not initiate new transactions across the link. For CMN-700:
  1. Poll the lnk<X>_ot_cbkwr bit in the por_ccg_ra_cxprtcl_link<X>_status register to make sure that it is cleared. This step ensures that there are no outstanding CopyBack requests targeting the link.

2. Take the link out of system coherency and DVM domains. For more information, see 4.4.7.7 Entry and exit protocol links from coherency domains and DVM domains on page 1150.

About this task

You must follow a specific process to bring down a protocol link between each CMN-700 CML gateway and the corresponding remote CML link that the gateway can communicate with. The term *link* is used here to refer to a protocol link. Multiple protocol links can be brought down at the same time by extending this sequence for each link.

CMN-700 CML gateways contain RA, HA, and LA nodes.

Procedure

1. Ensure that the link is up and can accept a new link down request by polling the following bits in the RA and HA protocol link control (*prtcl_link<X>_ctl) registers:
   - lnk<X>_link_up
   - lnk<X>_link_req

   These bits must be set before the link can accept a new link down request.
2. Clear the lnk<X>_link_req bits in the RA and HA protocol link control (*prtcl_link<X>_ctl) registers to make a link down request.
3. Poll the lnk<X>_link_ack bit in the RA and HA protocol link status (*prtcl_link<X>_status) registers to ensure that it is cleared.
   The hardware acknowledges a link down request by clearing the lnk<X>_link_ack. After a link down request is accepted, each side must stop granting local protocol credits and start returning remote protocol credits.
4. Poll the lnk<X>_link_down bits in the RA and HA protocol link status (*prtcl_link<X>_status) registers to ensure that each side has received all its protocol credits and is ready to go down.
   The hardware sets the lnk<X>_link_down bits to convey that it is ready for the link to go down.
5. Clear the lnk<X>_link_up bits in the RA and HA protocol link control (*prtcl_link<X>_ctl) registers to instruct both sides to deactivate the link.

Results

Link<X> is now down. No protocol message or credit transfers must occur across the link.

4.4.7.7 Entry and exit protocol links from coherency domains and DVM domains

CMN-700 CML gateway blocks, which contain RA, HA, and LA nodes, can establish a protocol link with a remote CML link. Each gateway block has software-programmable bits to allow the protocol links to enter and exit the system coherency domains and DVM domains.

The HA has 2 bits which facilitate snoop coherency domain entry and exit requests and acknowledgment:

- The lnk<X>_snoopdomain_req bit of the HA protocol link control (*prtcl_link<X>_ctl) register in each HA, HA controls snoop coherency domain requests for the link.

- The lnk<X>_snoopdomain_ack bit of the HA's protocol link status (*prtcl_link<X>_status) register, provides acknowledgment and status of the snoop coherency domain requests for the link.

The RA has 2 bits which facilitate DVM domain entry and exit requests and acknowledgment:

- The lnk<X>_dvmdomain_req bit in the RA's protocol link control (*prtcl_link<X>_ctl) register in each RA, controls DVM domain requests for the link.
- The lnk<X>_dvmdomain_ack bit in the RA's protocol link status (*prtcl_link<X>_status) register, provides acknowledgment and status of the DVM domain requests for the link.

See 3.3.8 RN entry to and exit from Snoop and DVM domains on page 98.

4.4.7.8 Program CML system to enable CXL communication

Use this procedure to enable CXL communication through a CCG block.

Before you begin

Before enabling CXL communication, you must first complete the bring up process. See #unique_949.

About this task

You can complete the steps in any order.

If the CMN-700 used in the host implementation and CCG is used to connect to a CXL Type3 memory device, then CCG must be enabled for CXSA mode. Enable CXSA mode in the CCG block by setting the cxsa_mode_en bit in RA's por_ccg_ra_cfg_ctl register.

Procedure

1. CXL Mode: Enable CXL mode by setting la_cxl_mode_en bit in LA's configuration control register (por_ccla_cfg_ctl).
2. CXL Host vs. Device configuration: If CMN-700 is used in the device implementation then set la_device_mode_en bit in LA's configuration control register (por_ccla_cfg_ctl). For host implementation this bit can be left to default value of 1'b0.
3. CXL connection Type (Type1/2/3):.
   RA connection Type: Program 2-bit cxl_type field in LA's configuration controlregister (por_ccla_cfg_ctl) to indicate the Type of CXL connection. By default this type field is set to 2'b11 (Type3).
4. CXL defined registers can be accessed either through CCG APB port or through CMN configuration access. To enable access through CCG APB port the following CCLA Aux_Ctl bit must be set "enable_cxl_regs_apb_access". By default this bit is cleared.
5. CXL.mem enable: CXL.mem enables is part of Flexbus port control DVSEC register (dvsec_flex_bus_port_control) and is expected to be set through CCG APB port. Alternatively the same bit can be set through CMN configuration access by writing the corresponding bits in por_ccla_dvsec_flex_bus_port_control register.
6. CXL Link Down: CCG-RA can give early completions, especially for write requests, so lnk0_ot_early_comp bit in RA's (por_ccg_ra_ccprtcl_link0_status) register bit must be polled

to make sure all the outstanding requests have received completions before a CXL device is disconnected (i.e. link down)

7. LDID Assignment for CXL Type 1 Device: Program unique LDID in HA's RAID to LDID LUT entry 0. See 4.4.7.3.4 Assign LDIDs to remote caching agents on page 1140

4.4.7.9 Programming optimization for DVMs and CPU events

Certain types of traffic like DVMs and CPU events can be optimized on CMN-700 mesh or CML link based on usage and configuration of a CML gateway block.


Note

This is for CML_SMP only.

The BTI, BTO, and Disable broadcast on VMF miss configuration registers described can be programmed to limit broadcast of non-sync DVMOps and DVMSyncs to Outer Sharable domains.

For (BTO,BTI) = (0,0) case, all DVMOps and DVMSyncs received by the DN will only be sent to RN-Fs, RN-Ds, and Peer DNs in the DN-of-interest's domain (inner sharable, only).

For (BTO,BTI)=(1,1) the SnpAttr field of non-sync DVMOp requests will be used determine if a particular request should be sent to a Outer Sharable domain.

Further, incoming DVMSync requests will only be shared to an Outer Sharable domain if a previous non-sync, Outer Sharable DVMOp request have been handled by the DN. Subsequent DVMSync requests will not be broadcast to Outer Sharable domains until the next non-sync, Outer Sharable DVMOp request is received and dispatched by the DN.

The Disable Broadcast on VMF Miss optimization will prevent the broadcast of a VMID-filterable non-sync DVMOp from being sent to the Outer Sharable domain.


Note

The `aux_ctl.disable_remote_broadcast_on_vmf_miss` may only be enabled if `aux_ctl.disable_vmf` = 0.To disable DVMOp and DVMSync optimization the following register programming must be set: `cfg_ctl.broadcast_dvmop_inner` = `1cfg_ctl.broadcast_dvmop_outer` = `1cfg_ctl.disable_remote_broadcast_on_vmf_miss` = 0

DN can be programmed to not send DVM snoops to a CML gateway block which is either configured for non-SMP connection or is an additional port in a port aggregation group (CPAG):

- Clear bit corresponding to the LogicalID of that CML gateway block in DN's por_dn_domain_cxra register

Additional port of a port aggregation group can be configured to not send CPU events across a CML link:

- Set lnk<x>_dis_cpuevent_prop in RA's protocol link control (* prtcl_link<x>_ctl) register

For more information about programming of DN registers for non-sync DVMOp and DVMSync optimization, see 3.9.4 DVM messages on page 230.

4.4.8 DT programming

You must follow several programming sequences to set up DTM watchpoints and DTC correctly.

For more information about the CMN-700 DT functionality, see 6. Debug trace and PMU on page 1200.

4.4.8.1 Program DTM watchpoint

Use this procedure to program watchpoint N, where N=0..3.

Procedure

1. Program the intended watchpoint matching fields by writing the appropriate values into the por_dtm_wpN_val and por_dtm_wpN_mask registers
   For example, source ID, target ID and opcode are possible matching fields.
2. Program the WP settings in the following register fields to select the device port and flit CHI channel:
   - wp_dev_sel2 and wp_dev_sel fields of the por_dtm_wpN_config register
   - wp_chn_sel field of the por_dtm_wpN_config register
3. Program the wp_grp field of the por_dtm_wpN_config register to select primary, secondary, or tertiary watchpoint group
4. Write 1 to the wp_combine field of the port_dtm_wpN_config register if two watchpoints must be combined

   
   Note

   The wp_combine field is present in WP0 and WP2 only.

5. Program the following register fields if trace packets are to be generated from this watchpoint:
   a) Program the wp_pkt_type field of the por_dtm_wpN_config register
   b) Write 1 to the wp_pkt_gen field of the por_dtm_wpN_config register
6. Write 1 to the wp_ctrig_en field of the por_dtm_wpN_config register if a cross trigger must be set up from this watchpoint
7. Program the wp_dbgtrig_en field of the por_dtm_wpN_config register if a debug watchpoint trigger must be set up from this watchpoint
8. Set the wp_cc_en field of the por_dtm_wpN_config = 1 if a cycle count is required in the trace packet
9. Write 1 to the trace_tag_enable field of the por_dtm_control register if a debug watchpoint trace tag must be generated
10. Write 1 to the dtm_enable field of the por_dtm_control register to enable the WP

4.4.8.2 Program DTC

Use this procedure to set up the CMN-700 debug trace control functionality.

About this task

The NIDEN input signal must be asserted for any trace and PMU operations. Before trace and PMU operations can occur, you must first program and enable watchpoint functions in the DTMs. The following registers and register bits are present only in the main DTC (DTC0):

- por_dt_secure_access register
- dt_en field of the por_dt_dtc_ctl register
- wait_for_trigger field of the por_dt_dtc_ctl register
- cc_start field of the por_dt_dtc_ctl register
- pmu_en field of the por_dt_pmcr register
- por_dt_pmsrr register
- ss_cfg_active field of the por_dt_pmssr register
- ss_pin_active field of the por_dt_pmssr register

Procedure

1. Write 1 to the dbgtrigger_en field of the por_dt_dtc_ctl register if DBGWATCHTRIG must be generated for DTM debug watchpoint trigger.
2. Write 1 to the atbtrigger_en field of the por_dt_dtc_ctl register if ATB trigger must be generated for DTM debug watchpoint trigger.
3. Write 1 to the cc_enable field of the por_dt_trace_control register to enable cycle count.
4. Write 0 to the dt_wait_for_trigger field of the por_dt_dtc_ctl register if no cross trigger is required.
5. Write 1 to the dt_en field of the por_dt_dtc_ctl register.

4.4.9 PMU system programming

You must follow specific programming sequences to set up the PMU, PMU snapshot, and PMU interrupt functionality correctly.

4.4.9.1 Set up PMU counters

Use this procedure to set up the PMU counters correctly.

Procedure

1. Ensure that the NIDEN input is asserted for any trace and PMU operations.
2. Program (dev)_pmu_event_sel register in the devices or XP.
3. Program the pmevcnt{0..3}_input_sel fields of the por_dtm_pmu_config register to select PMU event counter inputs.

The input can be from one of the following:

- A watchpoint.
- Selected events from the devices or XP, depending on step 2.

4. Program the following por_dtm_pmu_config register fields to select the paired top global PMU counters:
   - pmevcnt_paired
   - pmevcnt{0..3}_global_num
5. Program the pmevcnt{01, 23}_combined fields of the por_dtm_pmu_config for any combined local PMU counters.
6. Write 1 to the pmu_en field of the dtm_pmu_config register.
7. Program the cntcfg field of the por_dt_pmcr register to pair the 32-bit global counters to make a 64-bit counter.
8. Write 1 to the ovfl_intr_en field of the por_dt_pmcr register to enable interrupts on INTREQPMU on any global counter overflow.
9. Write 1 to the pmu_en field of the por_dt_pmcr register to start PMU operation.
10. Write 1 to the dt_en field of the por_dt_dtc_ctl register

4.4.9.2 Program PMU snapshot

Use this procedure to set up the PMU snapshot functionality.

Before you begin

The NIDEN input must be asserted for any trace and PMU operation.

About this task

For a system with multiple DTCs, the sub-DTC maintains snapshot status for the DTM within its own domain.

Procedure

1. Program PMU counters as described in 4.4.9.1 Set up PMU counters on page 1154.
2. Write 1 to the ss_req field of the por_dt_pmsrr register.
   This action causes the DTC to send a PMU snapshot instruction. On receiving this instruction, the DTM sends PMU snapshot packets to the DTC.

Results

The DTC updates the ss_status field of the por_dt_pmssr register after receiving PMU snapshot packets. Software can poll this register field to check if the snapshot process is complete.

por_dt_pmccntr is reset when snapshot happens.

4.4.9.3 Program PMU counter overflow interrupt

Use this procedure to set up the PMU counter overflow interrupt.

Before you begin

The NIDEN input must be asserted for any trace and PMU operation.

Procedure

1. Program PMU counters as described in 4.4.9.1 Set up PMU counters on page 1154.
2. Write 1 to the ovfl_intr_en field of the por_dt_pmcr register.
   Overflow of any PMU counter causes INTREQPMU to assert.
3. Write 1 to the ovfl_intr_en field in the all other por_dt_pmcr registers if your system has multiple DTCs.
4. Poll the pmovsr[7:0] field of the por_dt_pmovsr register when INTREQPMU is asserted to see which global counter causes the interrupt.
   For multiple DTCs, all por_dt_pmovsr registers must be polled.
5. Write 1 to the corresponding bit in the pmovsr_clr[7:0] field of the por_dt_pmovsr_clr register to clear INTREQPMU.

5. SLC memory system

This chapter describes the optional SLC memory system which is implemented by HN-Fs in the mesh.

5.1 About the SLC memory system

The SLC memory system consists of the HN-F protocol nodes in CMN-700.

There is a configurable number of instances (1-128) of the HN-F. Each HN-F node or slice has the following features:

- 0KB, 128KB, 256KB, 384KB, 512KB, 1MB, 1.5MB, 2MB, 3MB, or 4MB of SLC Data RAM and Tag RAM
- Combined *Point-of-Coherency* (PoC) and *Point-of-Serialization* (PoS)
- SF size based on SF_NUM_WAYS:
  - SF_NUM_WAYS = 16: 512KB, 1MB, 2MB, 4MB, 8MB
  - SF_NUM_WAYS = 28: 896KB, 1.8MB, 3.6MB, 7.2MB, 14.3MB
  - SF_NUM_WAYS = 32: 1MB, 2MB, 4MB, 8MB, 16MB
- SF Tag RAM configurable to have 16, 28 or 32 ways

Each HN-F in CMN-700 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 SLC
- Manages PoC and PoS functionality for ordering and coherency
- Tracks RN-F caching in the SF

The SLC memory system has the following features:

- *Physically Indexed and Physically Tagged* (PIPT)
- Coherency granule is a fixed length of 64B. SLC line size is a fixed length of 64B.
- Both SLC 16-way set-associative for 512KB, 1MB, 2MB, 4MB and 8MB sizes. 12-way for 3MB SLC configurations.
- Optionally, CMN-700 supports an *enhanced LRU* (eLRU) cache replacement policy that you can enable by setting a bit in the configuration register. eLRU is Dynamic Biased Replacement Policy. 2 bits per set, or way, track, and predict how soon a cache line is expected to be used again. This information is dynamically adjusted based on a few reference sets. By default, the SLC and SF victim selection policy is:
  - Pseudo random if all ways are valid
  - If there is an invalid way, it is not necessary to select a victim
  - Victim selection is required only if all ways are taken

- SLC and SF arrays:
  - Support one-cycle, two-cycle, or three-cycle non-pipelined tag array
  - Support two-cycle or three-cycle non-pipelined data array
  - SLC Tag, SF Tag, and SLC Data arrays are single-ported, supporting one read or write access with no concurrency available
  - SLC Tag, SF Tag, and SLC Data arrays are ECC SECDED protected, with inline ECC checking and correction
- 16, 24, 32, 48, 64, 80, 96, or 128-entry Address and data buffer, known as the *PoC Queue* (POCQ), to service:
  - All transactions from the CHI interface
  - SLC evictions to the memory controller
  - SF evictions and associated Write-Backs to the memory controller
- Fully configurable clustering of RN-Fs to support up to 512 caching agents tracked in SF
- CMO propagation to SN-F or SBSX:
  - Implements improved PCMO flow that is introduced in CHI-D
  - Conditional CMO propagation to the memory controller to support external DRAM caches
  - HN-F must be explicitly programmed, using the cmn_hns_sam_sn_properties register of the HN-F SAM, to enable this propagation to each SN-F
- Protocol flow control using programmable classes:
  - POCQ resources are allocated or rejected for protocol retry according to the class
  - POCQ resources are watermarked for different classes with user-configurable options
  - Class-based static grantee selection for CHI architecture credit return
  - See 5.5 HN-F class-based resource allocation and arbitration on page 1192
- Class-based request selection to memory controller
- Allocation in the SLC from snoop intervention. This feature enables data sharing through the SLC for multiple sharers.
- SLC state includes a caching LDID to detect dynamic read sharing
- Configurable 34-bit, 44-bit, 48-bit, or 52-bit *Physical Address* (PA) support
- PoC and PoS for all snoopable and non-snoopable, and cacheable and Non-cacheable transaction address space
- Supports ECC scrubbing for single-bit ECC errors on SF, Tag RAMs, SLC Tag RAMs, and SLC Data RAMs.
- Software-controlled error injection support to enable testing of software error handler routine
- Power management states to support:
  - Full powerdown of the SLC and SF, with HN-F only mode when both SLC and SF are powered down
  - Half the SLC ways powered down

  - Retention for SLC and SF
  - SLC full powerdown with SF on, when in SFONLY mode
- Arm TrustZone® technology support in SLC and SF
- Software-configurable (one, two, four, eight, or 12 ways) memory region locking support in the SLC
- Software-configurable (one, two, four, eight, or 12 ways) OCM support in the SLC
  - OCM memory does not require any physical memory backing
- CHI enhancements for:
  - *Direct Cache Transfer* (DCT)
  - *Direct Memory Transfer* (DMT)
  - Cache stashing
  - Atomics support
  - Data poison
  - Data parity (data check)
  - Trace tag
- Invisible SLC support:
  - CMN-700 HN-F implements an invisible cache. All accesses (cacheable, non-cacheable, and device types) are checked against the SLC and SF. The SLC cannot be cleaned and invalidated (flushed) by software using Arm architecture set, or way, operations. Software specific to CMN-700 is required to flush the SLC, as this Technical Reference Manual describes. Invisible SLC support eliminates the requirement to perform SLC flushes for software context switches from cacheable to non-cacheable.
- HNSAM supports:
  - Up to 64 non-hashed memory regions
  - Up to 8 hashed memory region (supporting two, four, or eight SN-F address hashing and SA aggregated hash functions)
  - Default regions, which support one, three, five, or six SN-F address hashing
- Supports CHI-D MPAM
- Supports CHI-E *Memory Tagging Extensions* (MTE)

5.2 SLC memory system components and configuration

CMN-700 *System Level Cache* (SLC) is a distributed, mostly exclusive, last-level cache that is implemented within the HN-F node.

When a sharing pattern is detected between RN-F clusters, the SLC is optimized to eliminate redundancy for private data lines from the RN-F. The SLC also enables redundancy, or pseudo-inclusion.

The SLC acts as DRAM cache for I/O coherent agents, that is, RN-Is. The SLC enables RN-Is to allocate or not allocate, according to the usage model.

The SF works with the SLC to track coherent lines that are present in the RN-F caches. The SF is fully inclusive of all the lines present in the RN-F caches. SF eviction invalidates the lines from RN-F caches to maintain this inclusion.

Usually, a particular coherent cache line is present only in the system-level cache or SF except when the line is shared between RN-F clusters. In the shared case, the line can be present in both the SLC and the SF.

5.2.1 HN-F configurable options

You can configure the HN-F in several ways.

The HN-F has the following configurable parameters:

- SF size based on SF_NUM_WAYS
  - SF_NUM_WAYS = 16: 512KB, 1MB, 2MB, 4MB, 8MB
  - SF_NUM_WAYS = 32: 1MB, 2MB, 4MB, 8MB, 16MB
- SF 16, 20, 24, 28, or 32 way set associate
- 16, 24, 32, 48, 64, 80, 96, or 128 POCQ entries. >64 configurations may have frequency implications
- One-cycle, two-cycle, or three-cycle Tag RAM arrays. For a given configuration, both SLC Tag and SF Tag have the same latency.
- Two-cycle or three-cycle Data RAMs, data, and SF array RAMs. All Data RAMs have the same latency.
- RN-F clustered mode and number of RN-Fs that are in each cluster using `SF_MAX_RNF_PER_CLUSTER`. `SF_MAX_RNF_PER_CLUSTER` supports values of 1, 2, 4, and 8.
- Number of extra bits in the SF RN-F tracking vector to allow for hybrid clustering of RNs using `SF_RN_ADD_VECTOR_WIDTH`. `SF_RN_ADD_VECTOR_WIDTH` has a minimum value of 0 and a maximum value of 127. The total number of bits in the SF RN vector must not exceed 128 bits. The total can be calculated as (`NUM_LOCAL_RNF` + `NUM_REMOTE_RNF`) / `SF_NUM_RNF_PER_CLUSTER` + `SF_RN_ADD_VECTOR_WIDTH`.

The HN-F has the following fixed parameter:

- HN-F CHI interface data-VC (DAT) width is 256 bits.

5.2.2 Snoop connectivity and control

Each HN-F can send three types of snoop.

The available types of snoop request are:

Directed

To one RN-F

Multicast

To more than one RN-F but not all. If SF clustered mode is enabled, multicast snoops might be more common.

Broadcast

To all RN-Fs.

When SF clustering is enabled, HN-F may utilize CHI-E SnpQuery opcode. It can be sent to an RN-F that requests MakeReadUnique to HN-F. This helps HN-F determine the cache line presence RN-F at the time of processing the transaction.

5.2.3 TrustZone technology support

The HN-F supports TrustZone® technology by treating the Non-secure bit from a request as part of the address.

TrustZone® enables the HN-F to treat Secure and Non-secure as two different areas of the memory space:

- The *Non-secure* (NS) bit is stored in the SLC and SF tags.
- Snoops also propagate the NS bit as part of the message.
- Any request to the memory controller also propagates the NS bit.

5.2.4 HN-F SAM configuration by SN type

CMN-700 supports multiple SN types. You must program the HN-F SAM according to the types of SN that the HN-F targets.

You can configure CMN-700 to use the following SN types:

- SBSX
- CHI-C, CHI-D, or CHI-E SN-F
- CXL.mem

To configure SN target types in the HN-F SAM, program the cmn_hns_sam_sn_properties register. For the description of this register, see 4.3.10.74 cmn_hns_sam_sn_properties on page 690.

The following table shows the bit values that you must program for each SN type.


Note

The value of <x> describes the specific SN target.

Table 5-1: cmn_hns_sam_sn_properties SN type values

SN type	<x>_sn_is_chic	<x>_sn_is_chie	<x>_sn_is_chif	<x>_sn_pcmosep_conv_to_pcmo
CHI-C SN-F	0b1	0b0	0b0	0b1

SN type	<x>_sn_is_chic	<x>_sn_is_chie	<x>_sn_is_chif	<x>_sn_pcmosep_conv_to_pcmo
CHI-D SN-F	0b0	0b0	0b0	0b0
CHI-E SN-F	0b0	0b1	0b0	0b0
SBSX	0b0	0b1	0b0	0b0


Note

- AXI memory controller does not support WR+PCMO, then you must program HN-F for this SN in CHI-D mode.
- If the AXI memory controller does not support PCMO on AW channel, then you must program HN-F as CHI-C SN mode.
- If the system uses CHI-E GrpIDExtn bits and the SN-F is CHI-B, CHI-C, or CHI-D, then you must set sn<x>_pcmo_conv_to_pcmo for HN-F to 0b1.
- If all SN types in the system do not support MTE, you must set cmn_hns_cfg.hnf_mte_mode_dis to 1.

5.2.5 Memory address decode error handling

HN-F SAM can detect Memory address decode error when an address is not programmed to target any SN.

For more information about the HN-F SAM, see 3.4.7 HN-F SAM on page 147

When HN-F encounters memory address decode error,

- The error is logged into HN-F RAS registers. See 3.8.5 HN-F error handling on page 216
- HN-F completes CHI transactions internally without sending any SN requests
- HN-F issues broadcast snoops to clean up upstream copies if needed
- Read Transactions:
  - If posion_on_mem_addr_dec_err_en is set, HN-F returns all 0s in data with poisons set; otherwise, HN-F returns all 1s in data without poisons.
- Write transactions:
  - Write data is dropped without allocating to SLC
  - Any SLC victims are dropped without writing back to SN


Note

HN-F ignores Memory address decode error if the address belongs to OCM region

5.2.6 Hardware-based cache flush engine

The HN-F supports a hardware-based cache flush engine mechanism to flush the SF and SLC. The flush engine ensures that all cache lines in the lower and upper range are flushed from the CMN-700 SF and SLC.

Various *Address-Based Flush* (ABF) configuration registers per HN-F instance support the cache flush engine:

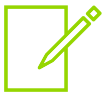
cmn_hns_abf_lo_addr
: ABF lower range address.

cmn_hns_abf_hi_addr
: ABF upper range address.

cmn_hns_abf_pr
: ABF Policy Register. Triggers flush start, indicates flush operation type.

cmn_hns_abf_sr
: ABF Status Register. Indicates flush completion and other status information.

The flush engine ensures that all cache lines in the lower and upper range are flushed from the CMN-700 SF and SLC. When all cache lines within this range are flushed, a bit in the cmn_hns_abf_sr register is set indicating that the flush engine has completed. If enabled, an interrupt, INTREQPPU, is then sent.


Note

The interrupt indication and complete bit in the cmn_hns_abr_sr registers is set regardless of normal completion or abort condition. To determine if a flush request completed normally or aborted, check the error bits in the cmn_hns_abf_sr register.

To complete the flush sequence, the HN-F performs the following steps:

1. Flush CMN-700 SFs. This operation flushes the lines in the lower-level caches. Lower-level write-backs go to memory and are not allocated to the CMN-700 SLC.
2. Flush the CMN-700 SLC.
3. On completion of the flush:
    a. The HN-F sets the status bit in the cmn_hns_abf_sr register when the flush is complete for that HN-F. If there are error conditions, they are also set in the cmn_hns_abf_sr register. This register is cleared when the next ABF request starts.
    b. The HN-F sends a completion message to the global Power/Clock/Reset unit, and an optional INTREQPPU is asserted when all HN-F instances have completed the flush.

ABF requests are processed in parallel to other ongoing requests from RNs. If an ABF request and another ongoing request target the same address, then no ordering or coherency guarantee is provided. Power management transition requests have higher precedence than ABF requests. An ABF request is only supported when the Power management state is in FAM, HAM, or SFONLY mode and the retention state is IDLE or RETENTION, not transitionary. While ABF is in progress,

any update to the *Power Policy Register* (PWPR) causes the ABF state machine to abort and the Power management request proceeds.

By default, the flush engine writes back any modified data to memory before invalidating the cache line from internal caches. Two more configuration modes are provided for user flexibility. Therefore, the flush engine has the following three modes of operation:

CleanInvalid Write back and invalidate. This mode is the default.
MakeInvalid In this mode, modified data is not written back to memory.
CleanShare In this mode, modified data is written back to memory but clean data remains in internal caches.

If *On Chip Memory* (OCM) is enabled and the address range overlaps with the ABF range, OCM behavior supersedes ABF. For SLC, this condition means that for CleanInvalid and CleanShare modes, no action is taken. For MakeInvalid, there is no difference in behavior, regardless of whether the address is in the OCM range or not. For SF flush, the behavior is the same between an OCM address match and not.

The following tables summarise SF and SLC caches for all three modes.

Table 5-2: SF cache operation

SF state	Hit		Miss	
ABF mode	SNP type to RN-F	Change SF state?	SNP to RN-F	Change SF state?
CleanInvalid	CleanInvalid	Yes	N/A	No
MakeInvalid	MakeInvalid	Yes	N/A	No
CleanShared	CleanShared	Yes	N/A	No

Table 5-3: SLC cache operation

SLC state		Modified		Exclusive or Shared		Invalid	
ABF mode	OCM match?	Evict line?	Change L3 state? (Final state)	Evict line?	Change L3 state? (Final state)	Evict line?	Change L3 state (Final state)
CleanInvalid	No	Yes	Yes (I)	No	Yes (I)	No	No (I)
	Yes	No	No (M)	No	No (ES)	No	No (I)
MakeInvalid	No	No	Yes (I)	No	Yes (I)	No	No (I)
	Yes	No	Yes (I)	No	Yes (I)	No	No (I)
CleanShared	No	Yes	Yes (E)	No	No (ES)	No	No (I)
	Yes	No	No (M)	No	No (ES)	No	No (I)


Note

The following assumptions are made:

- To ensure that coherency and ordering is maintained, RNs should not access a cache line within the flush range while ABF is in progress.
- Address ranges and trigger must be programmed for each HN-F in the SCG.

- Do not change ABF-related configuration register bits when the abf_enable bit of the cmn_hns_abf_pr register is set, until the abf_complete bit of the cmn_hns_abf_sr register indicates that the flush is done.
- HN-F must be in one of the three operational modes, FAM, HAM, or SFONLY. When flush starts, any update to the PWPR causes ABF to abort.
- SF must be enabled for the flush engine to operate. If SF is disabled, the flush engine aborts and indicates an error status in the cmn_hns_abf_sr register.
- When ABF completes, check the cmn_hns_abf_sr to ensure that ABF completed without any errors. If ABF aborted for any reason, then the cmn_hns_abf_sr indicates that the flush was aborted.

5.2.7 Software-configurable memory region locking

The HN-F supports variable-size memory regions that can be locked in the system-level cache with way reservation.

These variable-size memory regions ensure that locked lines are not evicted from the SLC, and any access to those lines is guaranteed to hit in the SLC. This guarantee applies to normal-memory cacheable and allocatable requests as described in CHI specifications. Non-allocating type requests might still be pushed to memory. The variable memory region is calculated as a factor of the total SLC size and number of ways that are locked.

For example, if the SLC is built with 16 ways, then way locking of 1, 2, 4, 8, or 12 yields 1/16, 2/16, 4/16, 8/16, or 12/16 of the SLC size respectively.

Software uses the following mechanism to program the HN-F configuration registers to enable region locking:

- The hnf_slc_lock_ways register specifies the total number of locked HN-F system level cache ways. This register can have 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_slc_lock_base0 register
  - hnf_slc_lock_base1 register
  - hnf_slc_lock_base2 register
  - hnf_slc_lock_base3 register
- A combination of the total SLC size, hnf_slc_lock_ways register, and the hnf_slc_lock_base0 register to hnf_slc_lock_base3 register defines the following:
  - The total amount of cache that is locked, calculated as follows:

Figure 5-1: Total cache locked equation

$$\frac{\text{Total SLC size x Number of locked ways}}{16}$$

    - 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_slc_lock_base0 to hnf_slc_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.
- The HN-F must be in the FAM power state. Memory Region Locking is not supported in other CMN-700 power states.


Note

The locked regions do not comprehend Secure versus Non-secure memory regions. If aliasing is performed between Secure and Non-secure regions, overlocking can occur.


Note

With Non-power of 2 hashing, where the number of HNFs within a cluster are non-power of 2, it might be necessary to over-provision locked/OCM ways. This might result in over-locking.


Note

SLC way locking is supported up to 128 HN-F in a single SCG.

The following tables specify various combinations of region size and the number of locked ways that software must program using the hnf_slc_lock_ways register and the hnf_slc_lock_base0 register to hnf_slc_lock_base3 register, based on an example configuration with an SCG total SLC size of 8MB.

Table 5-4: SLC region lock sizes

SLC size	Number of locked ways	Total locked region size	Locked ways	Number of ways per region	Region 0	Region 1	Region 2	Region 3
8MB	1	0.5MB	0	1	0.5MB	-	-	-
8MB	2	1MB	0-1	1, 1	0.5MB	0.5MB	-	-
8MB	4	2MB	0-3	1, 1, 1, 1	0.5MB	0.5MB	0.5MB	0.5MB
8MB	8	4MB	0-7	2, 2, 2, 2	1MB	1MB	1MB	1MB
8MB	12	6MB	0-11	2, 2, 4, 4	1MB	1MB	2MB	2MB

Table 5-5: Settings for hnf_slc_lock_baseX

Region size	Valid bits
0.5MB	[PA_WIDTH-1:19]
1MB	[PA_WIDTH-1:20]
2MB	[PA_WIDTH-1:21]
4MB	[PA_WIDTH-1:22]
8MB	[PA_WIDTH-1:23]

5.2.8 Software-configurable On-Chip Memory

The CMN-700 HN-F supports software configurable *On-Chip Memory* (OCM). This enables the creation of systems without physical DDR memory. It also allows a system to use SLC as scratchpad memory.

In OCM mode, the HN-F does not send requests to the SN-F. To enable OCM, the following requirements must be met:

- The HN-F must be in the FAM power state. Other CMN-700 power states are not supported in OCM mode.
- All OCM ways must be the same across all HN-Fs in a system cache group.

In OCM mode, the following CMOs terminate in the SLC:

- CleanInvalid and CleanShared CMOs terminate in the SLC without invalidation or performing a WriteBack to the SN-F.
- MakeInvalid invalidates the cache line in SLC, and can be used to invalidate the OCM region.

OCM mode can be enabled by programming the hnf_ocm_en bit in the cmn_hns_cfg_ctl 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 SLC size of the configuration when the hnf_ocm_allways_en is set to 1. If the hnf_ocm_allways_en bit is 0, region locking registers define the OCM regions. For more information about these region locking registers, see 5.2.7 Software-configurable memory region locking on page 1165.


Note

Region locking registers do not explicitly control Secure and Non-secure memory regions. Therefore, combined Secure and Non-secure memory regions must not exceed the total SLC size that is locked for OCM.

CMN-700 HN-F supports dynamic OCM mode entry and exit.

Restrictions:

- Cannot be used as a full OCM mode (hnf_ocm_allways_en must be 1'b0).
- Can only be used in 1, 2, 4, 8, or 12 way locking.
- If OCM all ways is enabled, then all incoming addresses map to OCM, even if the address is outside the SLC size range.
    - The OCM region copy would not function because the target memory range never goes to MC.

Instructions for entering OCM mode:

1. Stop all RN traffic to HN-Fs.
2. Transition HN-F from FAM to SFONLY operating mode as3.3.5 HN-F power domains on page 93 describes.
3. Enable OCM mode using config registers.
4. Transition HN-F from SFONLY to FAM operating mode.
5. Enable traffic to HN-F including to OCM region(s).

Instructions for exiting OCM mode:

1. Stop all traffic from RNs to OCM region(s).
2. Using system software, copy the SLC contents of the OCM address region to memory using new, seperate address space.
3. Initiate ABF in Makeinvalid mode to clear the OCM region(s) while OCM mode is still enabled..
4. When ABF is complete, disable OCM.

5.2.9 Source-based SLC cache partitioning

HN-F supports a feature to lock SLC ways for requesting nodes RN-F, RN-I, and RN-D.

This feature is an extension of the address-based way locking but the locked ways are based on the requestors instead of the programmed address. CMN-700 supports programming the number of ways that can be locked, RN devices for which these ways are locked, and allocation policies in each HN-F. With this feature enabled, the locked SLC ways are only available to the programmed RNs for any new cache line allocations.

You can enable the source-based way locking featured by programming the cmn_hns_rn_region_lock.rn_region_lock_en bit to `0b1` in each HN-F instance. You must also

explicitly enable the requesting nodes in the cmn_hns_rn*region_vec registers for which these ways are to be locked,. The requesting nodes are individually identified using the logical IDs or cluster IDs. See 5.2.14 Configuring clustered mode for SF tracking on page 1173. Each requesting node type RN-F, RN-I, and RN-D has different registers. They are uniquely identified in the CMN-700 system using logical IDs. The number of ways that are locked are programmed in the cmn_hns_slc_lock_ways.ways field.

In configuration with up to 32 RN-I and RN-D, each bit represents the RN-I or RN-D logical ID. If the configuration contains greater than 32 RN-I and RN-D but less than 64, then LSB bit of the logical ID is dropped when indexing the region lock registers. Similarly if the configuration contains more than 64 RN-I and RN-D, then the least significant 2 bits of logical ID are dropped when indexing into the region lock configuration registers.

The region locking feature has the following allocation modes:

Allocating new cache lines for matching RNs only in the locked ways

The matching RNs are restricted to allocate to the locked partition only. Enable this mode by setting the cmn_hns_rn_region_lock.rn_pick_locked_ways_only bit to `0b1`.

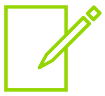
Allocating new cache lines for matching RNs to one of the locked or unlocked ways

This mode is the default behavior. In this mode, the locked ways are restricted only to the matching RNs but the unlocked ways are accessible by all the RNs.

Source-based SLC cache partitioning is supported only when *Enhanced LRU* (eLRU) mode is used.

Source-based cache partitioning is not supported:

- In other CMN-700 power states
- When the `CHI_MPAM_ENABLE` parameter is 'True'.


Note

The HN-F must be in the FAM power state.

5.2.10 Way-based SLC cache partitioning

You can partition each SLC cache instance into different regions. This partitioning enables each requesting node (RN-F, RN-I, RN-D) to allocate in one or more regions, each consisting of four consecutive ways.

Each region group has configuration registers that indicate which of the logical RN-F, RN-I, or RN-D managers can allocate to the corresponding group of SLC ways. By default, all RNs can allocate all 16 ways, as the following tables show.

Table 5-6: Logical RN-F ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-F ID								
				63	62	61	60	...	3	2	1	0
cmn_hns_slcway_partition0_rnf_vec	0xC48	[3:0]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition1_rnf_vec	0xC50	[7:4]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition2_rnf_vec	0xC58	[11:8]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition3_rnf_vec	0xC60	[15:12]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1

Table 5-7: Logical RN-I ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-I ID								
				31	30	29	28	...	3	2	1	0
cmn_hns_slcway_partition0_rni_vec	0xC68	[3:0]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition1_rni_vec	0xC70	[7:4]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition2_rni_vec	0xC78	[11:8]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition3_rni_vec	0xC80	[15:12]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1

Table 5-8: Logical RN-D ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-D ID								
				31	30	29	28	...	3	2	1	0
cmn_hns_slcway_partition0_rnd_vec	0xC88	[3:0]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition1_rnd_vec	0xC90	[7:4]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition2_rnd_vec	0xC98	[11:8]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition3_rnd_vec	0xCA0	[15:12]	{32'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1

The registers in these tables mask the ways that are available for an RN to allocate to at all times:

0b1 Indicates that the corresponding Logical RN ID or cluster ID can allocate in this region.
For more information about SF clustered mode and cluster IDs, see 5.2.14 Configuring clustered mode for SF tracking on page 1173.

0b0 Indicates that the corresponding Logical RN ID cannot allocate to this region.

To enable the way reservation only to a subset of RNs, the mask in the preceding registers must be programmed as the following example shows:

Example 5-1: Reserve ways 0-3 for RN-F {0-3}

1. Write `64'h000000000000000F` to cmn_hns_slcway_partition0_rnf_vec. This operation enables logical RN-F IDs 0, 1, 2, and 3 to allocate to ways 0-3. All other RN-F IDs (4-63) cannot allocate to these ways.
2. Write `32'h0` to cmn_hns_slcway_partition0_rni_vec. This operation disables all RN-Is from allocating to ways 0-3.

3. Write `32'h0` to cmn_hns_slcway_partition0_rnd_vec. This operation disables all RN-Ds from allocating to ways 0-3.


Note

The following conditions apply to this example:

- This feature cannot be used if region-based locking, source-based locking, or OCM is enabled.
- The region registers can be changed at runtime.
- When the way partitioning scheme is not being used, the preceding registers must be returned to the default values.
- Each RN must be configured to allocate to at least one partition. Setting the bit for a given RN to 0 in all partition registers defaults it to allocating in any partition.
- RN-I and RN-D each support a maximum logical ID of 96 using 32-bits in the partition mask register. In configurations with up to 32 RN-Is and RN-Ds, each bit represents the RN-I or RN-D logical ID. If the configuration contains greater than 32 RN-Is or RN-Ds but less than 64, then LSB bit of the logical ID is dropped when indexing the partition mask register. Similarly if the configuration contains more than 64 RN-Is and RN-Ds, then the least significant 2 bits of logical ID are dropped when indexing into the partition configuration register. In CML configurations,take care to assign LDID to remote RN-I and RN-Ds above the local RN-I and RN-D logical IDs. This requirement ensures that SLC partitioning is honored correctly across all RN-Is and RN-Ds.
- If SF clustered mode is enabled, HN-F uses the cluster ID instead of the full logical ID of the RNs. Therefore, all the RNs within a cluster can allocate to the locked ways.

Way-based SLC cache partitioning is supported only when *Enhanced LRU* (eLRU) mode is used.

Way-based SLC cache partitioning is not supported:

- In other CMN-700 power states
- When the `CHI_MPAM_ENABLE` parameter is 'True'.


Note

The HN-F must be in the FAM power state.

5.2.11 RN-F tracking in the SF

The CMN-700 HN-F SF tracks cache lines that come from specific RN-Fs using an RN-F vector index. Your CMN-700 configuration and the values of some configurable parameters determine the size of this width.

The maximum width of the vector index is 128 bits.

Use the following parameters to calculate the total width of the SF vector:

`SF_TOTAL_WIDTH = SF_MIN_WIDTH + SF_RN_ADD_VECTOR_WIDTH`

The configuration calculates `SF_MIN_WIDTH`, using the following equation:

`SF_MIN_WIDTH = ceil((NUM_LOCAL_RNF + NUM_REMOTE_RNF) / SF_MAX_RN_PER_CLUSTER)`

The `SF_MIN_WIDTH` calculation ensures that the SF vector index can track all the local and remote RN-Fs in your configuration. However, you can also configure the SF vector index at build time to contain extra bits by using the `SF_RN_ADD_VECTOR_WIDTH` parameter. This parameter lets you add extra bits to the SF vector, up to a combined maximum `SF_TOTAL_WIDTH` of 128 bits.

You can choose between two modes for tracking RN-Fs in the SF, or you can use a mixture of the modes. For more information about these modes, see 5.2.12 Non-clustered and clustered modes for SF RN-F tracking on page 1172.

5.2.12 Non-clustered and clustered modes for SF RN-F tracking

The CMN-700 SF uses non-clustered and clustered mode to track cache lines from RN-Fs. The mode that you use affects the number of RN-Fs that the SF can track.

Non-clustered mode

Each entry in the RN-F vector index is associated with a single RN-F. In this mode, CMN-700 is limited to a maximum of 128 RN-Fs. This limitation is because the SF can only track cache lines from a maximum of 128 RN-Fs. This limitation also applies to CML configurations.

Clustered mode

Each entry in the RN-F vector index is associated with a group of RN-Fs. You can group two, four, or eight RN-Fs into a cluster using this mode. This mode lets you increase the number of RN-Fs in a system up to a maximum of 512. The number of clusters is limited to 128 in this mode.

Both non-clustered and clustered mode use RN-F LDIDs to map RN-Fs to entries in the index. However, the way that the SF uses LDIDs to map RN-Fs to index entries differs between the two modes. In non-clustered mode, the whole LDID is used to map an RN-F to a single entry. In clustered mode, part of the LDID indicates which cluster group an RN-F belongs to and another part indicates the device within the group. In this case, the first part of the LDID is known as the cluster ID, and the second part is known as the device ID. Each bit in the SF vector represents the cluster ID of the RNs. For example, if there are four RN-Fs in a cluster, the SF is addressing all four RN-Fs. For any shared lines, HN-F snoops all four RN-Fs in the cluster.

For more information about the modes, including example configurations, see the following sections:

- 5.2.13 Configuring non-clustered RN-F tracking in HN-F SF on page 1173
- 5.2.14 Configuring clustered mode for SF tracking on page 1173
- 5.2.15 Identifying clusters and individual devices in clustered mode on page 1174

5.2.13 Configuring non-clustered RN-F tracking in HN-F SF

To enable non-clustered mode, set `SF_MAX_RNF_PER_CLUSTER` parameter to 1. The LDID of each RN-F in the system is assigned to a single index entry in the SF RN-F vector index.

For example, the following table shows an example of mapping between the RN-F vector index and the RN-F LDIDs in a configuration with 128 non-clustered RN-Fs.

Table 5-9: Example SF RN-F vector index for 128 non-clustered RN-F configuration

RN-F vector index value	RN-F LDID
0	LDID_0
1	LDID_1
2	LDID_2
...	...
127	LDID_127

In a CML configuration, all remote RNs must have LDIDs assigned to them after the local RN-F LDIDs have been assigned. For example, consider a system with 64 RN-Fs, divided into 16 local RN-Fs and 48 remote RN-Fs. In this system, LDIDs 0-15 are preassigned to the local RN-Fs. LDIDs 16-63 must then be assigned to the remote RN-Fs.

5.2.14 Configuring clustered mode for SF tracking

You can enable clustered mode for RN-F tracking in the SF by setting the `SF_MAX_RNF_PER_CLUSTER` parameter value to a valid value that is >1. This parameter specifies the maximum number of RN-Fs that are in a cluster group, up to a maximum of eight.

If clustered mode is enabled, there must be at least two cluster groups. You cannot cluster all RN-Fs into a single cluster group.

If there is a single RN-F in the cluster, then it must always use the lowest device ID in that cluster. Consider a four-way clustering, where the device ID fields are `0b00`, `0b01`, `0b10`, and `0b11`. In this configuration, you cannot use device ID values of `0b01`, `0b10`, or `0b11` unless `0b00` is also used.

You can combine local and remote RN-Fs into the same cluster. Each cluster can contain any number of RN-Fs, up to the value of the `SF_MAX_RNF_PER_CLUSTER` value. For example, in a system with `SF_MAX_RNF_PER_CLUSTER` = 4, cluster 0 could contain four RN-Fs and cluster 1 could contain only two RN-Fs.

The size of the SF RN-F vector index is partly configurable, as 5.2.11 RN-F tracking in the SF on page 1171 describes. You can use the SF_RN_ADD_VECTOR_WIDTH parameter to set up a mixed clustered and non-clustered tracking scheme. This type of scheme enables the SF to track some RN-Fs clustered in groups, and others individually.

When you choose to use clustered mode, also consider the following:

- SLC partitioning and SLC way locking using source ID use the cluster ID instead of the full LDID. Therefore, all RN-Fs in the cluster can use the partitioned ways for this source ID.
- Using clustered mode in the CMN-700 SF increases the likelihood of SF back invalidations if the number of sharers exceeds the maximum precise tracking ability of the SF vector. The following describes the precise mode tracking. Although unique cache lines are still precisely tracked in the SF vector index, any lines that go into shared mode can cause SF pollution. SF pollution can occur because the evict operations from the clustered RN-Fs do not clear the SF entry.

5.2.15 Identifying clusters and individual devices in clustered mode

For the SF to track and identify RN-Fs within cluster groups, the RN-F LDID bits are divided into separate components. These components are known as the cluster ID and the device ID.

The number of LDID bits that these components use depends on the following properties of your configuration:

- The number of RNs in the system that are addressable by the SF RN-F vector index. The width of the LDID scales up to a maximum size of 9 bits in a system with 512 RN-Fs.
- The value of the SF_MAX_RNF_PER_CLUSTER parameter.

Calculate the width of the device ID component of the LDID using the following equation:

Device ID width = $\log_2$(SF_MAX_RNF_PER_CLUSTER)

The device ID bits are mapped to the least significant bits of the LDID. The remaining bits of the LDID represent the cluster ID.

For example, the following table shows how LDID[8:0] is divided for different SF_MAX_RNF_PER_CLUSTER values in a 512 RN-F configuration in clustered mode.

Table 5-10: Cluster ID and device ID LDID components for 512 RN-F configuration

MAX_RNF_PER_CLUSTER value	LDID ranges	Clustering components
4	LDID[8:2]	Cluster ID[6:0]
	LDID[1:0]	Device ID[1:0]
8	LDID[8:3]	Cluster ID[5:0]
	LDID[2:0]	Device ID[2:0]

You must consider the cluster ID and device ID when assigning LDIDs to local and remote RN-Fs. Accounting for these values helps you ensure that RN-Fs are grouped in the cluster as required.

Example cluster mode configurations

The following table shows an example configuration with 512 RN-Fs, a 128-bit RN-F vector index, and cluster groups of four RN-Fs.

Table 5-11: Example clustered mode configuration with four RN-Fs per cluster group

RN-F vector index value (cluster ID)	Device ID 0 (LDID)	Device ID 1 (LDID)	Device ID 2 (LDID)	Device ID 3 (LDID)
0	LDID_0	LDID_1	LDID_2	LDID_3
1	LDID_4	LDID_5	LDID_6	LDID_7
2	LDID_8	LDID_9	LDID_10	LDID_11
...	...	...	...	...
127	LDID_508	LDID_509	LDID_510	LDID_511

The following table shows an example configuration with 512 RN-Fs, a 64-bit RN-F vector index, and cluster groups of eight RN-Fs.

Table 5-12: Example clustered configuration with eight RN-Fs per cluster group

RN-F vector index value (cluster ID)	Device ID 0 (LDID)	Device ID 1 (LDID)	Device ID 2 (LDID)	Device ID 3 (LDID)	Device ID 4...6	Device ID 7 (LDID)
0	LDID_0	LDID_1	LDID_2	LDID_3	...	LDID_7
1	LDID_8	LDID_9	LDID_10	LDID_11	...	LDID_15
2	LDID_16	LDID_17	LDID_18	LDID_19	...	LDID_23
...	...	...	...	...	...	...
63	LDID_504	LDID_505	LDID_506	LDID_507	...	LDID_511

You can use SF clustering in the following ways:

Fully precise mode

Each cluster only contains one RN-F. This mode behaves similar to non-clustered mode.

Hybrid mode

Each cluster can contain zero to `SF_MAX_RNF_PER_CLUSTER` RN-Fs

Fully clustered mode

Each cluster contains maximum RN-Fs up to `SF_MAX_RNF_PER_CLUSTER` parameter.

Consider a CMN-700 system with 8 RN-Fs and 4-way SF clustering is enabled in HN- F (`SF_MAX_RNF_PER_CLUSTER` = 4). The SF vector width is calculated as:

Minimum SF width = ((LOCAL_RNF+REMOTE_RNF)/`SF_MAX_RNF_PER_CLUSTER`)

The result of the calculation translates to SF tracking vector with 2 bits. A system builder can choose to have extra tracking bits in the SF using the `SF_RN_ADD_VECTOR_WIDTH` parameter. For example, if this parameter is set to 6, then the total SF tracking ability is extended to 8 clusters (TOTAL_SF_VEC_WIDTH = 8) with 4 RN-Fs each. Therefore, up to 32 RN-Fs can be tracked in the SF with LDIDs 0-31. In this configuration, the LDID_WIDTH = 5.

The following table shows the LDID mapping to cluster and device ID as per the SF tracking vector.

Table 5-13: LDID mapping to SF cluster

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	LDID2	LDID3
1	LDID4	LDID5	LDID6	LDID7
2	LDID8	LDID9	LDID10	LDID11
3	LDID12	LDID13	LDID14	LDID15
4	LDID16	LDID17	LDID18	LDID19
5	LDID20	LDID21	LDID22	LDID23
6	LDID24	LDID25	LDID26	LDID27
7	LDID28	LDID29	LDID30	LDID31

Each LDID entry in the preceding table contains RN-F attributes such as:

RN_Valid

Indicates a valid RN-F is mapped to this LDID.

***CML Port Aggregation* (CPA) enable**

If the RN-F is a remote RN-F, this attribute indicates that *CML Port Aggregation* (CPA) is enabled.

***CML Port Aggregation Group* (CPAG) ID**

If this RN-F is a remote RN-F, this attribute indicates the *CML Port Aggregation Group* (CPAG) ID.

Remote

To indicate this RN-F is a remote RN-F.

CHI protocol supported

The protocol version of this RN-F (CHI-B/C/D/E).

RN NodeID

Physical NodeID of this RN-F. For remote RN-Fs, this attribute can contain the CCG nodeID.

These fields are programmed in cmn_hns_rn_cluster[0-127]_physid_reg[0-3] register in HN-F. It is used for HN-F to send snoop requests.

Default LDID mapping to SF clustering table

Out of reset, MXP assigns sequential LDIDs to RN-Fs within each chip. If all RN-Fs are on the same chip, they are assigned LDID 0-7. You can program the attributes of each LDID as the following table shows. With the default LDIDs, only clusters 0 and 1 are used and clusters 2-7 are unused.

Table 5-14: Default LDID mapping to SF clusters

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	LDID2	LDID3
1	LDID4	LDID5	LDID6	LDID7
2	-	-	-	-

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7	-	-	-	-

In this mode, SF is tracking the RN-Fs at cluster level for shared cache lines. Therefore, tracking is imprecise and HN-F can snoop all the RN-Fs in that cluster when required.

Fully precise mode

Each cluster can be programmed to have one RN-F so that it can be precisely tracked in the SF. In this mode, default LDIDs in MXP must be programmed to match the cluster to LDID mapping the following table shows.

Table 5-15: Precise mode LDID mapping to clusters

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	-	-	-
1	LDID4	-	-	-
2	LDID8	-	-	-
3	LDID12	-	-	-
4	LDID16	-	-	-
5	LDID20	-	-	-
6	LDID24	-	-	-
7	LDID28	-	-	-

In this mode, every RN-F is precisely tracked in the SF vector.

Hybrid mode

In hybrid mode, the number of RN-Fs in each cluster can be different. In other words, the SF tracks RN-Fs with a mix of precise and imprecise tracking. You must program the default LDIDs in MXP to match the following table.

Table 5-16: Hybrid LDID mapping to SF clusters

SF tracking vector ID (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	-	-
1	LDID4	-	LDID6	-
2	LDID8	-	-	-
3	LDID12	-	-	-
4	LDID16	-	-	-
5	LDID20	-	-	-

SF tracking vector ID (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
6	-	-	-	-
7	-	-	-	-

In this mode, HN-F tracks cluster 0 and 1 in imprecise mode for shared cache lines. Clusters 2-5 are tracked in precise mode.

Fully clustered mode

In fully clustered mode, every cluster contains as many valid RN-Fs as when configured using the `SF_MAX_RNF_PER_CLUSTER` parameter. For example, in a system with 32 RN-Fs and `SF_MAX_RNF_PER_CLUSTER` = 4, the SF width = 8. Therefore, the LDID to cluster mapping is fully populated.

Table 5-17: Fully clustered mode mapping of LDID to SF clusters

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	LDID2	LDID3
1	LDID4	LDID5	LDID6	LDID7
2	LDID8	LDID9	LDID10	LDID11
3	LDID12	LDID13	LDID14	LDID15
4	LDID16	LDID17	LDID18	LDID19
5	LDID20	LDID21	LDID22	LDID23
6	LDID24	LDID25	LDID26	LDID27
7	LDID28	LDID29	LDID30	LDID31

In this mode, shared cache lines are tracked at cluster level. HN-F can snoop all the RN-Fs in the cluster when required to resolve coherency.

For the 8-cluster examples, the following registers must be programmed for each LDID and they must be consistent within each chip.

HN-F registers:

For an example of 8 clusters with 4 RN-Fs each, you must program the following HN-F registers:

- cmn_hns_rn_cluster0_physid_reg[0-1]
- cmn_hns_rn_cluster1_physid_reg[0-1]
- cmn_hns_rn_cluster2_physid_reg[0-1]
- cmn_hns_rn_cluster3_physid_reg[0-1]
- cmn_hns_rn_cluster4_physid_reg[0-1]
- cmn_hns_rn_cluster5_physid_reg[0-1]
- cmn_hns_rn_cluster6_physid_reg[0-1]
- cmn_hns_rn_cluster7_physid_reg[0-1]

These registers contain attributes for each RN-F such as its NODEID, SrcType, CPA info, remote or local, and a valid bit.

MXP registers:

Default LDID assignment of RN-Fs attached to each MXP device port p[0-3] (directly or behind CAL) are captured in the register:

- por_mxp_device_ port_connect_ldid_info_p[0-3]: Read-only register

In hybrid mode and precise modes, if you change the default LDID, you must program the MXP LDID override registers appropriately:

- por_mxp_p[0-3]_ldid_override
  - Program these registers to override the LDID assignment of RN-F or RN-Fs attached (directly or behind CAL) to each MXP device port p[0-3].
  - Reset value of these registers is the default LDID assignment.
  - These registers are present only if clustering is enabled and RN-F or RN-Fs are connected to this device port.

RA register:

Assign *Requesting Agent IDs* (RAIDs) for local RNs.
For each local RN, identified by their LDID, program RAIDs and set the corresponding valid bit in the following registers:

- por_ccg_ra_rnf_ldid_to_exp_raid_reg [0-127]
- por_ccg_ra_rni_ldid_to exp_raid_reg [0-9]
- por_ ccg_ra_rnd_ldid_to_exp_raid_reg [0-9]

This programming sets up the LDID to RAID LUT.

LDID override. Consider a situation where the LDID of a given RN-F is overridden by programming the corresponding register in MXP. In this case, the following registers at RA must be programmed with the overridden value for that particular LDID:

- por_ccg_ra_rnf_ldid_to_ovrd_ldid_reg [0-127]

HA registers:

Program unique LDIDs for each remote caching agent in the following register:

- por_ccg_ha_rnf_exp_raid_to_ldid_reg [0-255]

Set the ldid<X>_rnf bit, which marks the remote agent as a caching agent, and set the respective valid bit.

The LDID values for remote RN-Fs must be greater than those values that are used by the local RN-F nodes, unless RN-Fs LDIDs are overridden. See RA and MXP sections for more details on LDID override.

HN-F also supports precise tracking up to n-sharers when clustering is enabled. This mode is only available when `TOTAL_SF_VEC_WIDTH` >= 2*(LDID_WIDTH+1). That is, the SF RN-F tracking vector

must have enough bits to track at least 2 LDIDs and a valid bit for each LDID. In the preceding example with 8 clusters and 4-way clustering, LDID_WIDTH is 5 bits wide. Therefore, precise mode is not available in this configuration.

If `SF_RN_ADD_VECTOR_WIDTH` = 4 in the preceding example, then precise mode is enabled, and it can track 2 sharers without having to snoop all RN-Fs within a cluster. This mode is enabled by default when a configuration has enough bits to track at least 2 RN-Fs. Disable this mode by programming the cmn_hns_cfg_ctl.sf_rnf_vec_precise_mode_en config bit to 1'b0. In the preceding example with 8 clusters and `SF_RN_ADD_VECTOR_WIDTH` = 4, the RN-F tracking vector in SF is 12 bits wide. In the precise tracking mode, the RN-F vector represents the decoded RN-F LDIDs as the following table shows.

Table 5-18: SF vector tracking RN-Fs in Precise mode

11	10	9	8	7	6	5	4	3	2	1	0
Second RN-F LDID[4:0]					Valid	First RN-F LDID[4:0]					Valid

When a cacheline is accessed the first time, the LDID of the requesting RN-F is tracked in the RN-F tracking vector with valid set to 1. Then, if a second RN-F accesses this same cache line, it occupies the second entry in the preceding table. Even if these 2 RN-Fs belong to separate clusters with 4 RN-Fs each, HN-F tracks them precisely. For example, consider a situation where the first RN-F is from cluster 2 and the second RN-F is from cluster 7. In this case, the preceding decode LDID vector fully tracks the true LDID of the requestors. In this mode, any evict operation from these RN-Fs is also precise and so there is no SF pollution.

Then consider a situation in which a third RN-F from cluster 9 accesses the same cache line. In this case, the SF switches from precise to imprecise mode because it does not have enough bits to track the third RN-F in precise mode. The RNFVEC is updated as the following table shows.

Table 5-19: Vector tracking RN-Fs in imprecise mode

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	0	0	0	0	1	0	0

The imprecise mode is reset when HN-F receives either:

- A request that requires the cache line to migrate to RN-F, such as ReadUnique/ MakeReadUnique
- Writes that require SF entry to be cleared

5.3 Error reporting and software-configured error injection

HN-F detects and reports several types of errors to the error block.

HN-F supports the following types of errors:

Correctable errors

For example, single-bit ECC error detection and correction in the SLC Tag RAM, SF Tag RAM, and SLC Data RAM

Deferred errors

For example, double-bit ECC error detection in SLC Data RAM

Uncorrectable errors

For example, double-bit ECC errors in the SLC or SF Tag RAMs

If the DATACHECK_EN parameter is enabled, HN-F can also support Data parity error detection in the SLC Data RAM. These errors are logged as Deferred Errors.

For logging and reporting all error types, HN-F follows the procedures described in 3.8 Reliability, Availability, and Serviceability on page 208.

For information regarding the error source, see the ERRSRC field of 4.3.10.27 cmn_hns_errmisc on page 640.

5.3.1 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 SLC double-bit ECC data errors.

The HN-F configuration register for a particular logical thread enables configurable error injection and reporting.

Any Cacheable read for which the HN-F provides the data is defined as a system cache hit. If error injection and reporting are enabled, any system cache hit drives the subordinate error from the system cache pipe and a fault interrupt through the RAS control block for that read. This functionality emulates a double-bit ECC error in the SLC data RAM but does not pollute the SLC data RAM through the fill path.


Note

This mechanism is designed to mimic SLC data ECC errors for SLC hits. If enabled, this mechanism only causes an error to be logged and optionally an interrupt to be generated. SLC misses do not drive any subordinate errors or error interrupts. Error injection on SLC hits does not alter the Resp*, Poison, or Data fields in the DAT flit that is returned to the RN.

For more information about configuring error injection, see 4.3.10.28 cmn_hns_err_inj on page 642.

5.3.2 Software-configurable parity error injection

The HN-F supports software-configurable parity error injection.

This feature enables testing of the software error handler routine for parity error. For more information about enabling this feature, see 4.3.10.29 cmn_hns_byte_par_err_inj on page 643. This register specifies the byte lane from 0-31 in which a parity error is introduced. The memory data is uncorrupted with such injection, but the Data Check field of the DAT flit that is returned to an RN is altered.

5.4 Transaction handling in SLC memory system

The CMN-700 SLC memory system handles various types of CHI operations and transaction fields. The structure of the overall system and how each component is configured affects how these transactions are handled.

5.4.1 Cache maintenance operations

CMN-700 uses several CHI *Cache Maintenance Operations* (CMOs).

The following operations are supported:

- CleanInvalid
- CleanShared
- MakeInvalid
- CleanSharedPersist
- CleanSharedPersistSep

These operations always look up the SLC and the SF, and take the following actions:

- Clean and invalidate the line if present in the SLC.
- If the CMO is address hits in the SF, the HN-F sends a snoop to the RN-F post SF lookup if necessary.
- If the cache line is modified in the SLC or in the cache of the RN-Fs, the HN-F initiates a memory controller WriteBack if necessary.


Note

If the CMO is MakeInvalid, there is no WriteBack to the memory controller.

The SF does not track RN-F coherence while the HN-F is in NOSFSLC state. Therefore, the RN-F caches must be flushed before transitioning from NOSFSLC to SFONLY, HAM, or FAM states.

5.4.2 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
- ReadPreferUnique

- MakeReadUnique
- ReadNotSharedDirty

The HN-F also supports system monitor functionality for Non-cacheable exclusive support. For more information about exclusives, see the *AMBA® 5 CHI Architecture Specification*.


Remember

Each HN-F in CMN-700 can support tracking of up to 512 logical processors for non-Cacheable exclusive operations:

- In configurations with up to 64 RN-Fs, HN-F supports 64 exclusive monitors.
- In configurations with 64-144 RN-Fs, HN-F supports 144 exclusive monitors.
- In configurations above 144 RN-Fs, the total number of exclusive monitors is equivalent to the total number of local and remote RN-Fs, RN-Is, and RN-Ds, up to a maximum of 512 exclusive monitors.

For Non-cacheable exclusive, the system programmer must ensure that there are no more logical processors capable of concurrently sending exclusive operations than the number of exclusive monitors.

Cacheable exclusive is implemented based on Snoop Filter and not constrained by the number of exclusive monitors.

5.4.3 DataSource handling

CMN-700 populates the DataSource field of a CompData response to specify the source of the data.

DataSource information can be used:

- To determine the usefulness of a PrefetchTarget (Memory controller prefetch) transaction
- To profile and debug software to evaluate and optimize data sharing patterns

Table 5-20: DataSource encodings

Source of data	Message	Encoding
HN-I	Default (non-memory source)	0b0000
RN-F	Peer processor cache within local cluster	0b0001
RN-F	Local cluster cache	0b0010
HN-F	*System Level Cache* (SLC)	0b0011
RN-F	Peer cluster cache	0b0100
Remote chip	Remote chip caches	0b0101
SN-F or SBSX	PrefetchTarget was useful	0b0110
SN-F or SBSX	PrefetchTarget was not useful	0b0111
RN-F	Local cluster cache, unused prefetch	0b1010
HN-F	System Level Cache (SLC), unused prefetch	0b1011

CMN-700 drives the DataSource value only when the source of the data is either HN-F or HN-I. For other data sources, CMN-700 acts as a conduit.

The encoding that is used by CMN-700 to indicate a data source is the same as the suggested value in the *AMBA® 5 CHI Architecture Specification*. Any deviation from the specified encodings might result in unexpected behavior.

5.4.4 CMO and PCMO propagation from HN-F to SN-F or SBSX

CMN-700 supports propagation of CMO and PCMO requests for a given cache line to the memory controller. The completion point of these requests is programmable.

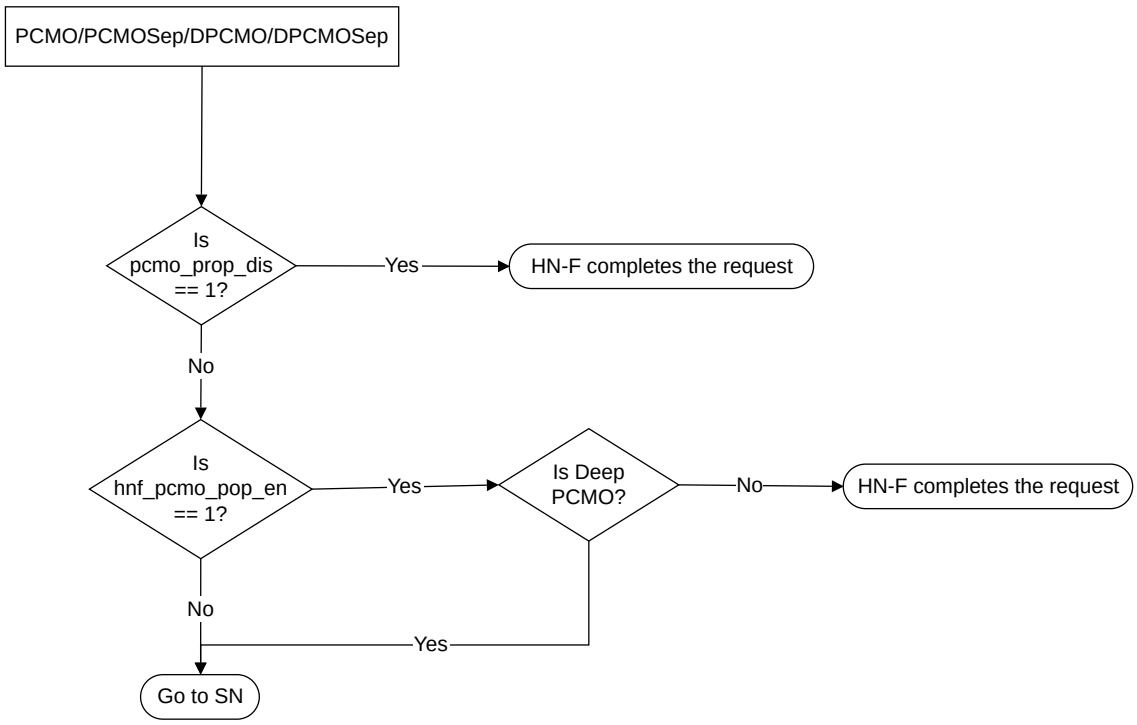
This feature ensures that the cache line has been written to the memory controller and that any copies in the CMN-700 system have been removed. Conditional CMO and PCMO propagation to the memory controller also supports external DRAM caches. You can enable or disable this feature in the cmn_hns_sam_sn_properties register bits for each HN-F corresponding to each SN-F. For SBSX with an AXI4 subordinate memory device, you must disable CMO and PCMO propagation in HN-F.

The following HN-F SAM configuration register bits decide the CleanSharedPersist (PCMO) and CleanSharedPersistSep (PCMOSep) completion point:

- cmn_hns_sam_sn_properties.pcmo_prop_dis
- cmn_hns_cfg_ctl.hnf_pcmo_pop_en

You can program this behavior per SN. Deep is an attribute in the CHI-D request flit and applies to PCMO type requests (DPCMOs). HN-Fs use the Deep attribute and the HN-F SAM configuration bits to decide the PCMO request completion point. The following diagram shows the PCMO and PCMOSep request completion point decision process.

Figure 5-2: PCMO and PCMOSep completion point flow diagram



If an HN-F is programmed to propagate PCMO requests to an SN-F, and the SN-F does not support PCMOSep requests, you must set the cmn_hns_sam_sn_properties.X_sn_pcmosep_conv_to_pcmo bit to `0b1`. This setting enables the HN-F to complete the request by converting PCMOSep type requests to PCMO. The HN-F generates a Persist response to the requestor on receiving a completion of PCMO response from SN-F.

Warning

CHI-C and CHI-D SN-Fs do not support the GroupIDExt field in the REQ flit. Therefore, if both of the following criteria are true for your CMN-700 system, you must program HN-Fs to terminate persistent CMOs:

- CMN-700 contains any CHI-C or CHI-D SN-Fs.
- RNs in the system can issue persistent CMOs.

Warning

If the `SBSX_CMO_ON_AW` parameter = 0, you must program the cmn_hns_sam_sn_properties.X_sn_pcmosep_conv_to_pcmo configuration bit. Setting this bit prevents HN-F from sending PCMO opcodes that map to two-part PCMOs on AW to SBSX.

You can also program HN-Fs to propagate CleanShared, CleanInvalid, and MakeInvalid CMOs to an SN-F. The following table shows how to program this behavior using the cmn_hns_sam_sn_properties register.

Table 5-21: CMO propagation programming in HN-F SAM

HN-F SAM attribute (cmo_prop_en)	CMO completion point
0b0	HN-F
0b1	SN

See the following register descriptions for more information:

- 4.3.10.74 cmn_hns_sam_sn_properties on page 690
- 4.3.10.6 cmn_hns_cfg_ctl on page 606

5.4.5 Memory System Performance Resource Partitioning and Monitoring

CMN-700 supports *Memory System Performance Resource Partitioning and Monitoring* (MPAM). MPAM features enable software to optimize the use of memory resources and to monitor how those resources are used.

If MPAM is enabled, then an extra MPAM field is added to the REQ and SNP channels. This field must be stored and propagated to the downstream target.

For more information about MPAM, see the following documents:

- *AMBA® 5 CHI Architecture Specification*
- *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A*

5.4.5.1 MPAM propagation

When MPAM is enabled, CMN-700 propagates the MPAM fields throughout the network. Various nodes are designed to propagate the MPAM field when processing requests.

The following node types propagate the MPAM field:

- MXP
- RN-I
- HN-F and HN-I
- CML RA and CML HA
- SBSX

The AXMPAM_EN parameter controls if MPAM bits are stored and propagated on RN-I, HN-I, and SBSX bridges.

If MPAM support is required in CMN-700, set the CHI_MPAM_ENABLE parameter.

When an HN-F node receives a request, it stores the details from the MPAM field. If the request misses in the SLC, then the HN-F must drive the stored MPAM values onto the outgoing request to the memory controller. The MPAM field contents are stored in the `SLC_TAG` array and propagated into requests that are sent to memory.

When MPAM is enabled, an extra register field, *_mpam_override_en, is added to the MXP. You can program this field to override the MPAM value of the RN-F request with the set value in the register. See 4.3.13.15 por_mxp_p0-5_mpam_override on page 896.

When MPAM is enabled, the RN-I has the following extra components:

- Two extra signals on the AXI or ACE-Lite subordinate port, ARMPAM[10:0] and AWMPAM[10:0]
- An extra register field, *_mpam_override_en, which you can program to override the request MPAM value with the set value in the register


Note

If AXMPAM_EN = 0, then the MPAM override is active by default.

The CML RA node type has the following MPAM modes:

SMP mode

The CML RA passes the MPAM field on the USER field of the request. When snooped, the CML RA receives the MPAM field and passes it through the CHI SNP MPAM field.

Non-SMP mode

The CML RA drops the MPAM field that is received on the CHI request. The CML RA also does not receive MPAM field on CML snoops in this mode.

CXSA mode

The CML RA passes the MPAM field on the USER field of the request, even though CXSA is in non-SMP mode. You can use a configuration bit to enable passing of MPAM attributes when in CXSA mode.

The CML HA node type has the following MPAM modes:

SMP mode

The CML HA receives MPAM fields through the USER field of the request and passes them through the CHI MPAM field. On incoming CHI transactions, the CML HA passes CHI MPAM values through on the USER field.

Non-SMP mode

The CML HA drops the MPAM values that it receives on CHI snoop. The CML HA does not receive the MPAM attributes on CML requests.

If `AXMPAM_EN` = 1, the HN-I and SBSX node types propagate CHI.RXREQ.MPAM onto the AXI pins. If AXMPAM_EN = 0, the HN-I and SBSX node types drive 0s onto AXI pins.

In CMN-700, MPAM support is applicable to the SLC.

5.4.5.2 MPAM configuration

CMN-700 provides several configuration parameters to configure MPAM features that are used by the interconnect. The MPAM feature ID register, cmn_hns_mpam_idr, provides information on the MPAM features that are supported in the design.

The number of partitions and the number of performance monitoring groups that are supported is configurable at build time. The default number of partitions is 64 for Non-secure partitions and 16 for Secure partitions. The default number of performance monitoring groups is two.

For more information about the configuration options, see 2.5.2 Mesh sizing and top-level configuration on page 25 and 2.5.3 Device placement and configuration on page 30.

For more information about supported MPAM features, see 4.3.11.3 cmn_hns_mpam_idr on page 801.

For more information about the software-programmable MPAM override mechanism, see 5.4.5.6 Software-programmable MPAM override on page 1190.

5.4.5.3 Cache portion and capacity partitioning

MPAM support in the interconnect is for the CMN-700 SLC. MPAM is supported only with an *enhanced Least Recently Used* (eLRU) cache replacement policy. When MPAM is enabled, eLRU mode is enabled automatically when either cache portion or cache capacity partitioning are enabled.

The SLC supports cache portion partitioning that is based on the following masks:

MPAMCFG_CPBM

The CPBM value for a request PARTID determines which cache portions a request can allocate.

MPAMCFG_CMAX

The CMAX value for a request PARTID determines the percentage of the SLC that a request can use.

Cache portion and capacity partitioning have the following features:

- The number of portions is the same as the number of ways in SLC.
- The hnf_mpam_ccap_idr_cmax_wd field of the cmn_hns_mpam_ccap_idr register is set to 7. This setting provides granularity of 0.78% ($1 / 2^7$) SLC for cache capacity partitioning.
- In HAM mode, portions 15:8 are aliased to portions 7:0. Cache capacity is adjusted for half the cache.
- CMN-700 supports address-based locking, including OCM, with MPAM. Locked ways are not available for MPAM-based partitioning. Cache capacity is adjusted to account for locked ways.


Note

CMN-700 does not support source-based or way-based SLC partitioning with MPAM.

If using way locking with MPAM, you must program the lock registers first. Then, to determine the number of available portions, read the hnf_mpam_cpor_idr_cpbm_wd field of the cmn_hns_mpam_cpor_idr register. Locked ways also reduce cache capacity.

HN-F MPAM counter values are not accurate when exiting retention state and can result in underflow conditions.

5.4.5.4 Cache capacity monitoring

The following describes cache capacity monitoring.

MPAM provides a mechanism for monitoring SLC usage:

- The HN-F `MPAM_NUM_CSUMON` parameter determines how many monitors are supported.
- The cmn_hns_X_msmon_cfg_csu_flt and cmn_hns_X_msmon_cfg_csu_ctl registers determine filter and control for each monitor.


Note

The interconnect has two banks for these registers, S and NS, denoted by X in the register or register field name.

The hnf_X_msmon_cfg_csu_ctl_capt_evnt field of the cmn_hns_X_msmon_cfg_csu_ctl register supports external capture events 6 and 7:

- External capture event 6 is triggered using PMUSNAPSHOT interface.
- External capture event 7 is triggered as the *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A* describes.


Note

Multiple capture events cannot be triggered within 32 cycles of each other.

5.4.5.5 MPAM error logging and reporting

CMN-700 implements programmable registers that can enable, disable, and modify MPAM error logging and reporting behavior.

The MPAM error status register, cmn_hns_X_mpam_esr, and the MPAM error control register, cmn_hns_X_mpam_ecr, define MPAM-related error logging.


Note

The interconnect has two banks for these registers, S and NS, denoted by X in the register or register field name.

You can enable interrupt generation for MPAM-related errors by setting the hnf_X_mpam_ecr_inten field of the cmn_hns_X_mpam_ecr register to `0b1`. If interrupt generation is enabled, level-sensitive interrupts, INTREQMPAMERRS or INTREQMPAMERRNS, are triggered for defined error cases.


Note

MPAM error reporting has the following exceptions:

- When SLC size is 0K, no errors are detected or reported.
- REQ PARTID or PMG out- of-range errors are not detected or reported when:
  - HN-F is in SFONLY mode.
  - MPAM features are disabled (by configuring the auxiliary control register).

For more information about MPAM errors, see *Error conditions in accessing memory-mapped registers* in the *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A*.

See the following register descriptions for more information:

- 4.3.11.15 cmn_hns_ns_mpam_ecr on page 814
- 4.3.11.16 cmn_hns_ns_mpam_esr on page 815
- 4.3.12.17 cmn_hns_s_mpam_ecr on page 855
- 4.3.12.18 cmn_hns_s_mpam_esr on page 856

5.4.5.6 Software-programmable MPAM override

CMN-700 has registers in various nodes to let software override the MPAM values generated by those nodes. The behavior of the override mechanism depends on the system configuration and security constraints.

The following nodes support a software override mechanism:

- RN-I
- RN-D
- CML-HA
- MXP

RN-Is, RN-Ds, and CML-HAs contain programmable registers to override the MPAM values for their own requests. The MXP contains programmable registers to override MPAM values for requests that RN-Fs generate.

The MXP software MPAM override mechanism is useful in the following situations:

- RN-F is a device that is compliant with CHI-D or CHI-E, and therefore supports MPAM, but MPAM is disabled in that device
- RN-F is a device that does not support MPAM

In these situations, the MXP MPAM override mechanism allows these devices to take part in the MPAM tracking scheme within the interconnect.

Each MXP contains an MPAM override register for each device port, por_mxp_p0_mpam_override, por_mxp_p1_mpam_override, por_mxp_p2_mpam_override, and por_mxp_p3_mpam_override.

Software can configure these registers to generate an MPAM override value for request flits that are uploaded on that device port by RN-Fs. These registers are applicable only when an RN-F is connected to the port.

For other device types, these registers are redundant and not used.

The following table shows the format of the override registers.

Table 5-22: por_mxp_p{0,1,2,3}_mpam_override register format

[63:25]	[24]	[23:17]	[16:8]	[7:5]	[4]	[3:1]	[0]
Reserved	REQ MPAM PMG	Reserved	REQ MPAM PartID	Reserved	REQ MPAM NS	Reserved	REQ MPAM override enable

The override enable bit indicates that the contents of the registers are valid, and therefore must be set when the other fields have the intended values.

The MXP MPAM override feature has the following security requirement:

By default, the MPAM.NS subfield has the same value as the NS field of the REQ flit. When the override enable bit is set, the NS override field drives the NS field of the REQ flit.

The override behavior in the MXP depends on the CHI version that the RN-F supports and whether the enable bit is set as the following table shows.

Table 5-23: MXP override behavior

CHI_MPAM_ENABLE setting	RN-F CHI version	Override enable bit set or not set	Override behavior
1	CHI-B or CHI-C	Not set	All MPAM fields in the REQ flit, except the MPAM NS field, are driven to the default value, which is `0x00`
1	CHI-B or CHI-C	Set	All fields, including the MPAM NS field, are overridden
1	CHI-D	Not set	The request uses the original MPAM values of the REQ flit
1	CHI-D	Set	All MPAM fields are overridden with the contents of the register


Note

When the RN-F CHI version is CHI-E, it is not necessary for MPAM NS and Request NS to match. For example, you can use Request NS = 0 and MPAM NS =1.

5.4.6 MTE support in HN-F

CMN-700 HN-F supports CHI-E *Memory Tagging Extensions* (MTE). For all coherent and non-coherent operations, if requests come with MTE opcodes, HN-F services them by fetching the required tags from SN-F.

In accordance with the CHI-E architecture, HN-F also flushes the dirty tags downstream to SN-F when required.

HN-F also supports finishing match operation in HN-F itself. By programming the hns_mte_no_sn_match in hns_cfg_ctl register, the match operation would always be done in HN-F and not sent downstream to SN-F.

HN-F supports to propagate clean the MTE tag to SN when the cacheline is victimized from SLC, by programming slc_victim_clean_tag_transfer_en to 1 in cmn_hns_cfg_ctl register.

To disable the MTE handling in HN-F when the software is running without MTE opcodes, program the hnf_mte_mode_dis bit of the cmn_hns_cfg_ctl register to 1. In this mode, HN-F ignores any requests for tags and drops dirty tags. For MTE Match requests, HN-F synthesizes a dummy Tag Match response to complete the protocol flow.


Note

MTE is not supported for address regions belonging to CXL attached memory

5.5 HN-F class-based resource allocation and arbitration

CMN-700 POCQ supports class-based resource allocation and arbitration for increased flexability and control.

5.5.1 Class assignment

QoS decoding happens inside the HN-F.

A request from RN is assigned a class based on QoS or Request Type (Opcode) of a dynamic, not previously retired, request. The cmn_hns_class_ctl config register provides configurability to

determine this class assignment. The assigned class is then used for POCQ resource allocation and arbitration to SLC/SF access. It is also used when arbitrating for TXREQ to an SN.

QoS based class assignment

cmn_hns_pocq_qos_class_ctl provides configurability to assign a class based on QoS min/max values.


Note

All QoS values must be assigned one, and only one, class.

Request Type (Opcode) based class assignment

Incoming request opcode can be classified as follows:

Class 0

Read type requests

Class 1

Copy Back requests, including copyback wr+cmo

Class 2

Non Copy Back requests, including non-copyback wr+cmo

Class 3

All other requests

5.5.2 POCQ resource allocation

POCQ entries are allocated based on three configurable registers per class. These registers define different limits for each class.

cmn_hns_pocq_alloc_class_dedicated

Dedicated entries are reserved entries for a given class in POCQ. For a given class, if the number of requests in POCQ are less than the dedicated entries, then any incoming request to that class is not retried. Unused dedicated entries belonging to a class are unavailable for other classes to use.

cmn_hns_pocq_alloc_class_max_allowed

max_allowed indicates the maximum number of entries allowed for any given class in POCQ. When max_allowed is reached for a given class, any incoming request for that class is retried, even if POCQ has available entries.

cmn_hns_pocq_alloc_class_contended_min

contended_min indicates the required number of entries for a class under contention. A class is under contention when POCQ is full and requests are retried. These entries are not reserved. Instead, requests for a class below its contended minimum are given higher priority than ones above its contended minimum for credit grant arbitration.

The contended minimum value for a class is used for prioritizing classes for static credit grants after requests have been retried. When static credits are available and all classes have met or exceeded their number of dedicated entries, the number of active entries in the POCQ for each class is compared against the class's contended_min value. Classes whose number of active POCQ entries are below their contended_min value are prioritized for static credit grants.


Note

The total number of dedicated entries, including SEQ reserved for snoop filter evictions, must not exceed the number of POCQ entries.

For any given class, Arm recommends that:

- max_allowed > contended_min > dedicated


Note

max_allowed must not be less than the dedicated entries.

Retried requests are arbitrated per class to provide static credit grant back to requesting agent by the following priority order, highest to lowest:

- Request classes below dedicated with round robin arbitration across classes
- Request classes below contended min with round robin arbitration across classes
- Request classes below max_allowed in weighted round robin across classes

Weights for weighted round robin can be configured in “cmn_hns_class_retry_weight_ctl”. Within a given class, static grants are allocated to different RNs based on round robin arbitration.

The following figures show possible POCQ resource allocation for various classes for dedicated and max_allowed. Dedicated is a minimum for a class and max_allowed is a maximum per class.

Figure 5-3: POCQ possible dedicated

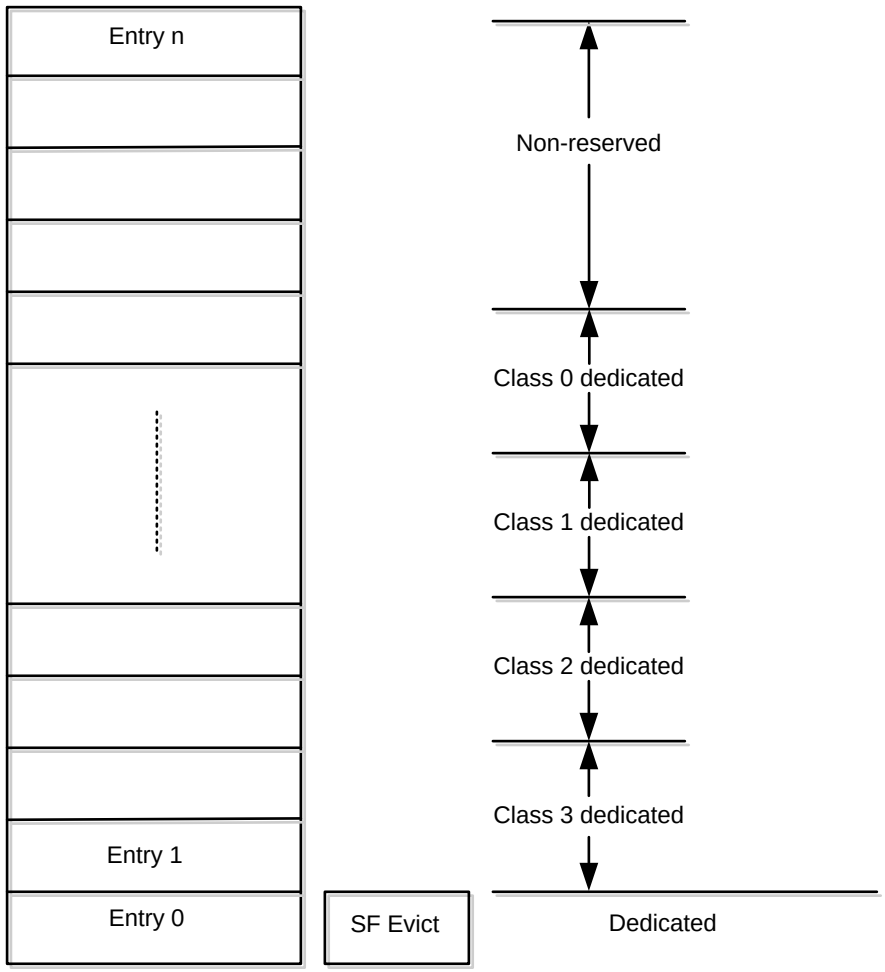
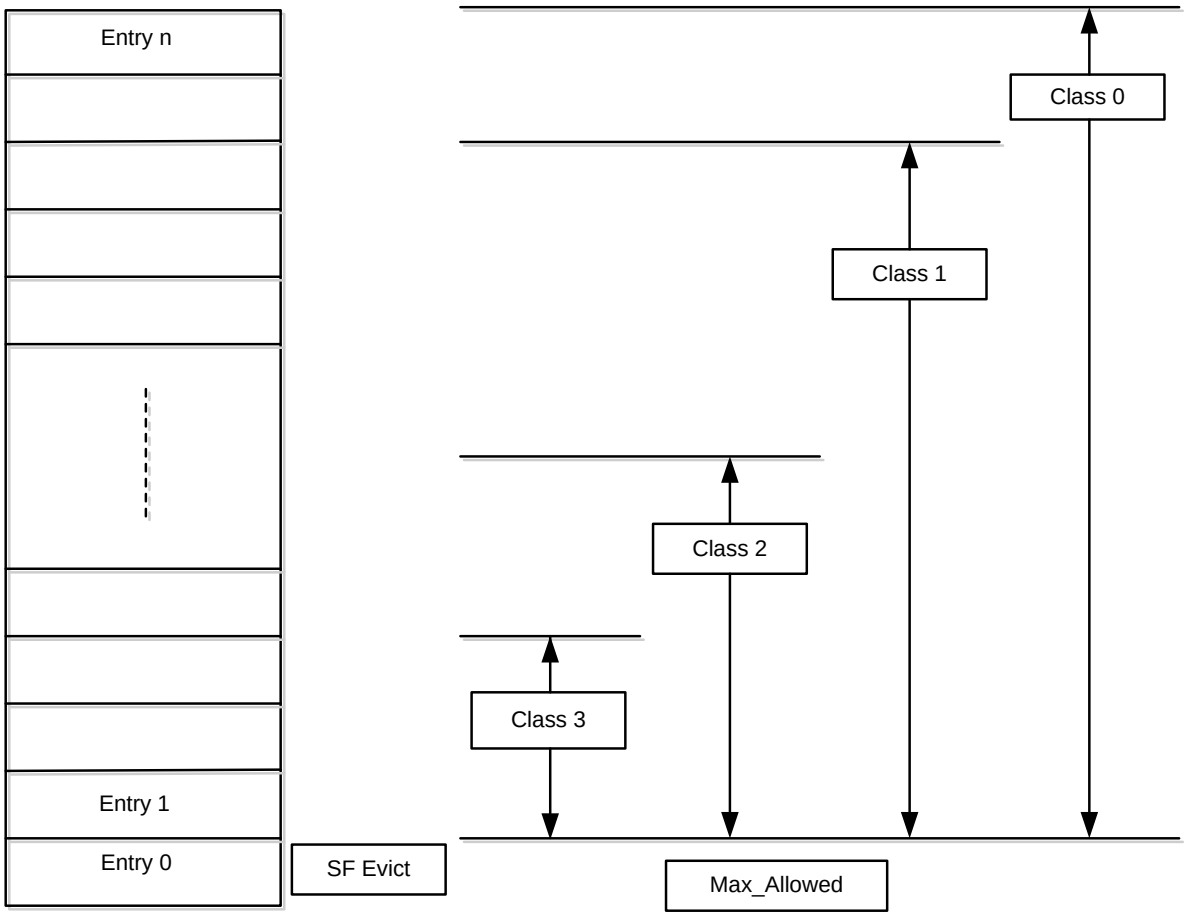


Figure 5-4: POCQ Default max_allocated



See 3.11.2.4 HN-F QoS support on page 248 for the QoS class and POCQ resource availability

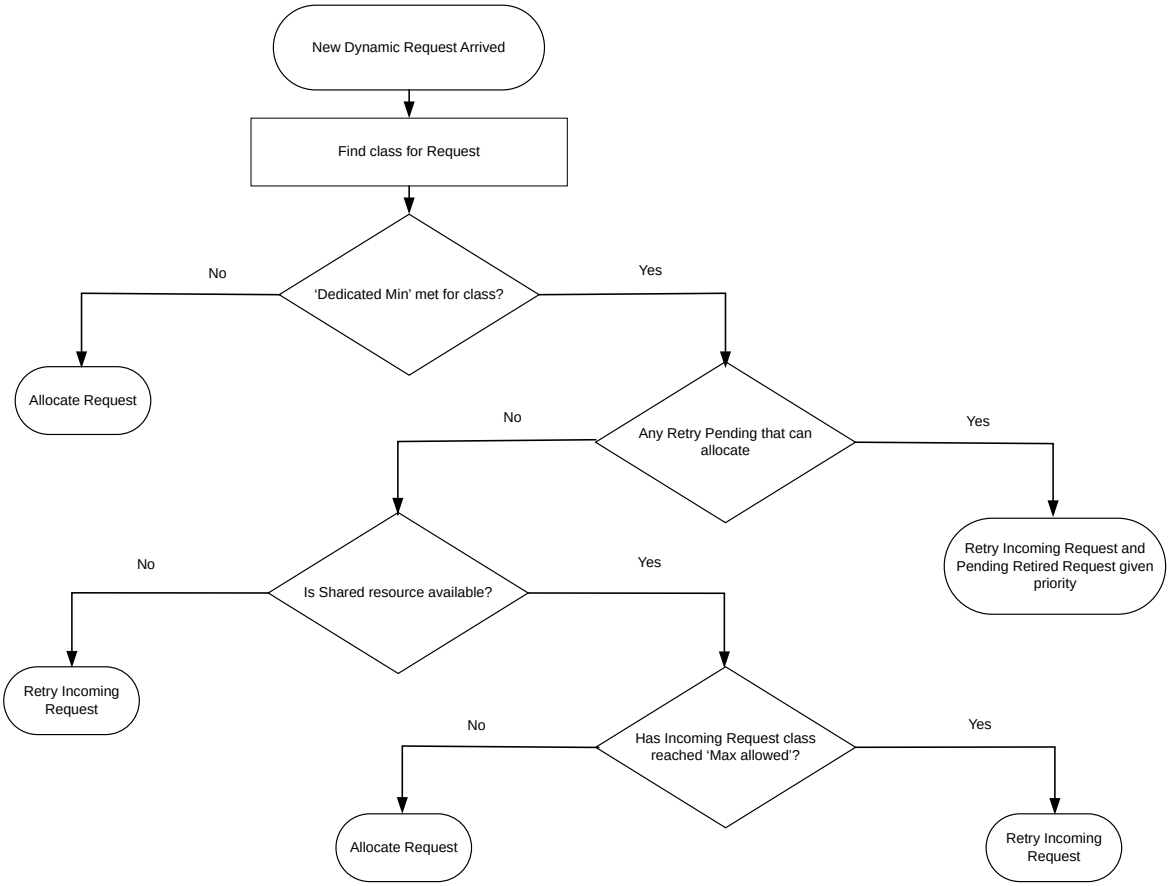
5.5.2.1 Request retry based on POCQ resource allocation

The following figure shows the decision making process to allocate an incoming dynamic request to POCQ or to retry.

Note

An incoming static request cannot be retried so it is allocated in POCQ.

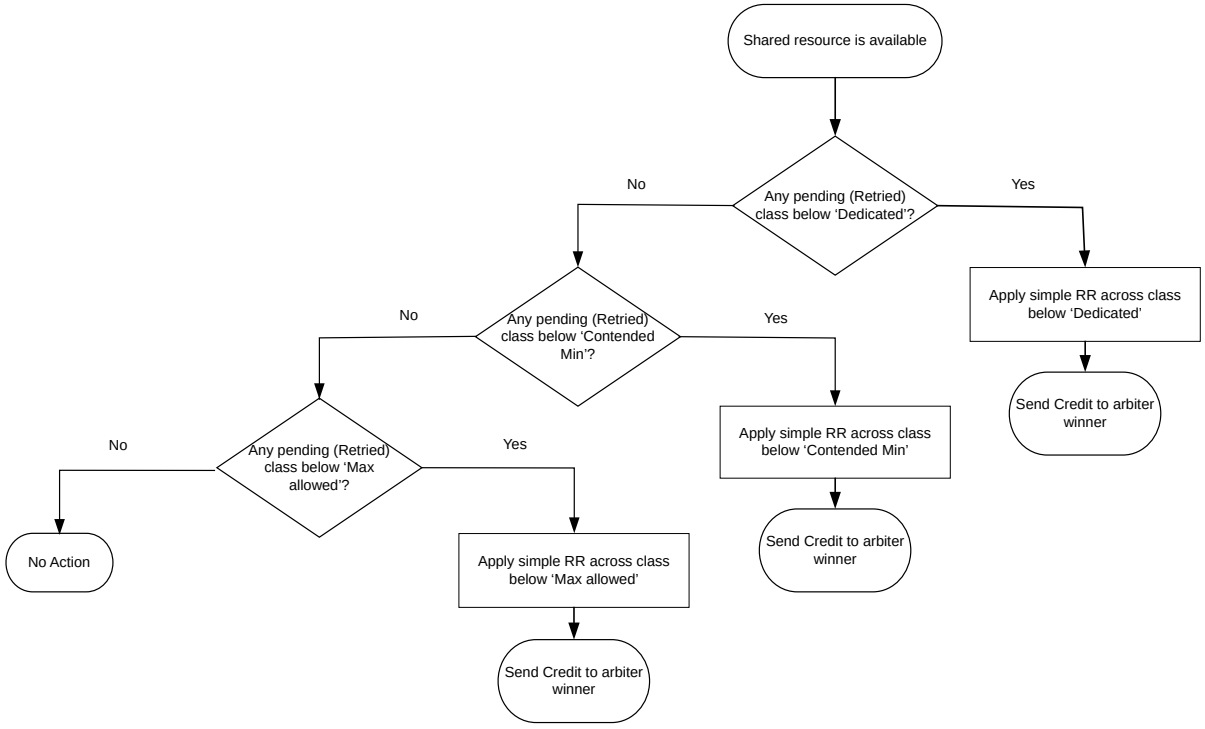
Figure 5-5: Request retry based on POCQ resource allocation



5.5.2.2 Credit grant for retried request

The following figure shows how credit is granted for different class requests when a request is retried. Within any given class, simple *Round Robin* (RR) arbitration is used.

Figure 5-6: Credit grant for retried request



5.5.3 POCQ request arbitration

Requests from POCQ to the SLC/SF pipe are selected by the following priority order, highest to lowest:

- Oldest requests in POCQ: ready for SLC/SF access
- Lookup requests: weighted RR per class
- Any eligible requests: weighted RR per class

Requests from POCQ to SN are selected by the following priority order, highest to lowest:

- Oldest requests in POCQ: ready to be sent to SN
- Static request: RR per class
- All pending SN requests: weighted RR per class

Configure the weights for weighted RR in cmn_hns_class_pocq_arb_weight_ctl.. The same weights are used for SLC, SF and SN weighted RR.

6. Debug trace and PMU

This chapter describes the *Debug Trace* (DT) and *Performance Monitoring Unit* (PMU) features that CMN-700 implements.

6.1 Debug Trace system overview

CMN-700 provides at-speed self-hosted *Debug Trace* (DT) capabilities.

The CMN-700 DT capabilities include:

- Watchpoint-initiated and trace-tag-initiated transaction tracing
- Globally synchronized cycle counters for precise tracing
- CHI trace tag generation
- CoreSight™ ATB trace streaming
- Access to trace data through configuration registers
- Cross trigger support
- Secure debug support
- Event-based interrupts

The CMN-700 DT system consists of a set of *Debug Trace Controllers* (DTCs) and *Debug Trace Monitors* (DTMs) distributed across the interconnect. DTCs are located inside HN-Ds and HN-Ts while DTMs are located inside XPs.

All DTCs, including the manager DTC, have an ATB interface and the following signals:

- DBGWATCHTRIGREQ
- DBGWATCHTRIGACK
- INTREQPMU

The following signals are only present in the manager DTC:

- NIDEN
- SPNIDEN
- PMUSNAPSHOTREQ
- PMUSNAPSHOTACK

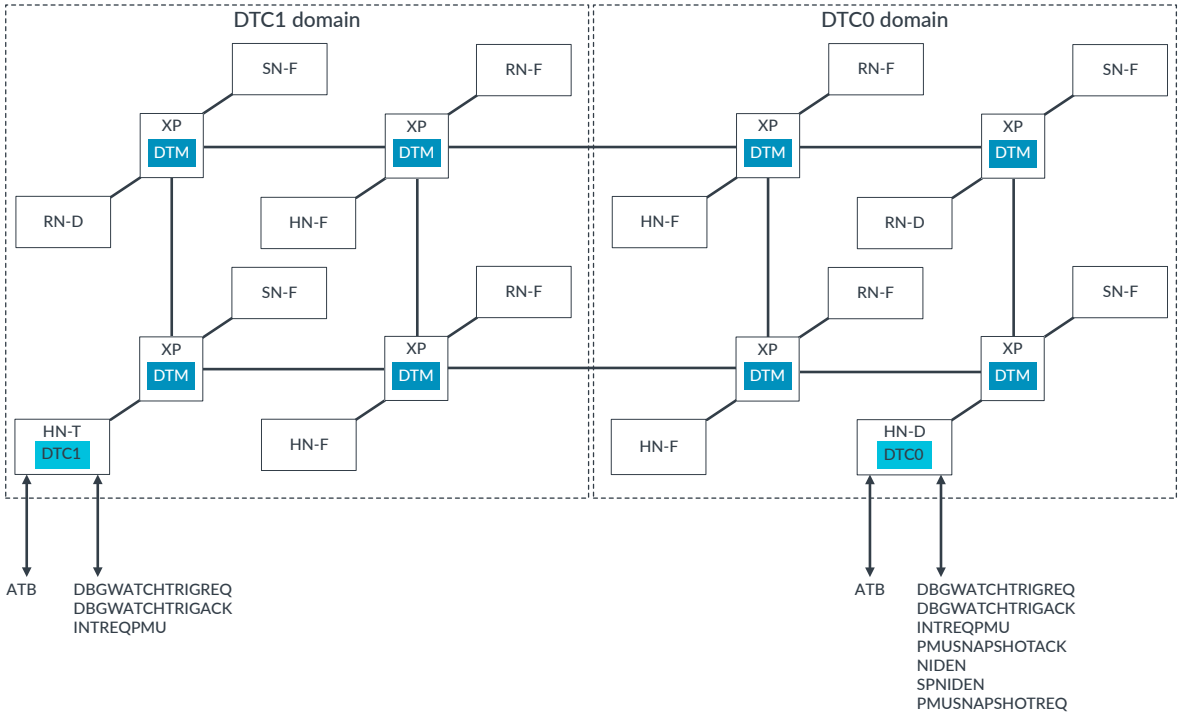

Note

NIDEN and SPNIDEN are propagated from the manager DTC to all DTMs. When asserted, they must remain asserted for at least 72 clock cycles. Likewise, when deasserted, they must remain deasserted for at least 72 clock cycles. This

requirement ensures that all internal CMN-700 components transit into their debug and trace states correctly.

The following figure shows an example DT system with two DTC domains.

Figure 6-1: Example DT System with two DTC domains



Arm recommends one DTC domain per 16 XPs. A combined maximum of 63 XP are permitted in a single DTC domain. For a system where the number of XPs is greater than 63, at least one HN-T node is required for each set of 63 XPs.

For example, a system with 96 XPs would require at least one HN-T node, while a system with 144 XPs would require at least two HN-T nodes.

Each DTC is associated with a set of DTMs to form an exclusive DTC domain. When enabled, DTMs within a DTC domain collect trace data and transmit it to the associated DTC. The number of HN-T nodes in the mesh determines the number of DTC domains.

In a DT system comprising multiple DTCs, the DTC that is located inside the HN-D is designated as the manager DTC or DTC0. You assign DTMs to DTCs by configuring XP parameters within Socrates™ IP Tooling platform.

Each DTC domain must be built using contiguous XPs.

The DT system implements the following functions:

- Monitoring CHI flits at XP device ports using four sets of *WatchPoints* (WPs) in each DTM

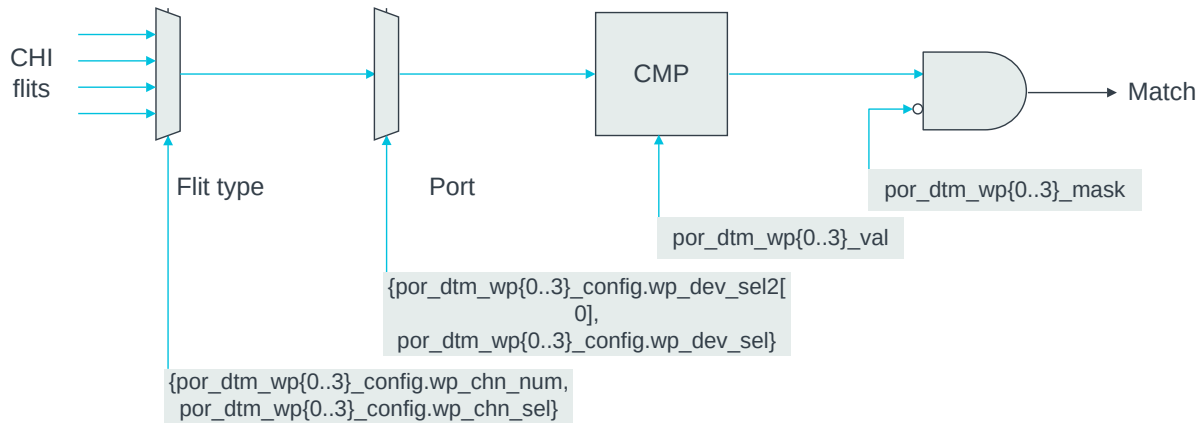
- Flit trace generation and storage at each DTM with control register access to trace packets
- Trace tag generation
- Debug trigger signaling and trace packet streaming over the *ATB* at each DTC
- Internal event-based cross trigger generation and broadcast to all DTMs
- Globally synchronized cycle counters

6.1.1 DTM watchpoint

A DTM has four *WatchPoints* (WPs) that monitor flit uploads and downloads at XP device ports.

WPs monitor flits by matching on a subset of flit fields that you specify using a pair of VAL and mask registers. The following figure shows the WP comparator and the registers that control this functionality.

Figure 6-2: DTM WP comparator



The port selection fields, wp_dev_sel2 and wp_dev_sel behave differently depending on the `MXP_MULTIPLE_DTM_EN` parameter value for the mesh when an XP has more than two device ports. The concatenated value {wp_dev_sel2, wp_dev_sel} selects which XP device port is monitored by the watchpoint.

When `MXP_MULTIPLE_DTM_EN` is FALSE, the XP will contain a single DTM and {wp_dev_sel2,wp_dev_sel} will take values from 2'b00 to 2'b11 to select one of the up to four device ports 0-3.

When `MXP_MULTIPLE_DTM_EN` is TRUE, the XP will contain two DTMs and each DTM will monitor at most two device ports. In this case, {wp_dev_sel2, wp_dev_sel} each DTM will only use values 2'b00 and 2'b01, to select between ports 0 or 1, and ports 2 or 3,

respectively.

A WP can be configured to monitor flits from up to 4 XP device ports and one of four CHI channels:

- REQ
- RSP
- SNP
- DAT

In addition, the WP can be configured to do one or more of the following tasks on detecting a flit match:

- Set trace tag bit on the flit.
- Generate flit trace.
- Generate cross trigger to DTC.
- Generate debug trigger to DTC.
- Increment PMU counters.


Note

Two WPs within a group can be combined for complex matching. For example, WP0 and WP1 can be combined, just as WP2 and WP3 can be combined. The four DTM WPs are assigned to flit uploads and downloads according to the following groups:

- WP0 and WP1 are assigned to flit uploads.
- WP2 and WP3 are assigned to flit downloads.

See 4.4.8.1 Program DTM watchpoint on page 1153 for the DTM watchpoint programming sequence.

6.1.1.1 WP match value and mask register

The WP flit matching criteria are specified using a 64-bit match value register, por_dtm_wpN_val, and a 64-bit mask register, por_dtm_wpN_mask. These registers allow matching of up to 64 bits of the flit.

N = 0, 1, 2, or 3 for the match value and mask registers.

To specify the value for matching, write the value into the por_dtm_wpN_val register. The value of the por_dtm_wpN_mask register specifies the bits that must be masked from the match comparison, and therefore ignored. To specify that a bit must be masked, write a 1 into the corresponding bit position in the por_dtm_wpN_mask register.

The CHI flit fields are divided into the following match groups:

REQ channel
: Primary, secondary, and tertiary match groups.

RSP channel
: Primary match group.

SNP channel
: Primary and secondary match group.

DAT channel

Primary and secondary match group.

The following tables specify the flit fields that belong to each of the groups for the different CHI channels.

Flit matching from two different match groups requires two WPs to be combined. If the fields from the different match groups uploaded on the REQ channel are to be matched, then WP0 and WP1 must be combined. WP0 and WP1, are assigned one of the uploaded fields from the different match groups. If the fields from the different match groups downloaded on the REQ channel are to be matched, then WP2 and WP3 must be combined. WP2 and WP3, are assigned one of the uploaded fields from the different match groups.


Note

SRCID and TGTID fields in the match groups will match against either the SrcID field or the TgtID field of a flit, depending on which watchpoint is used. Watchpoints 0 and 1 (upload direction) match on the TgtID field, while watchpoints 2 and 3 (download direction) match on the SrcID field.

The following table shows REQ channel width and bit ranges for the primary match group.

Table 6-1: REQ channel: primary match group

Field	Width	Bit range
SRCID/TGTID	11	10:0
STASHNID/RETURNNID/SLCREPHINT[6:0]	11	21:11
StashTgtValid/Endian/Deep	1	22:22
{StashLPValid, StashLP[4:0]}	6	28:23
OPCODE	7	35:29
SIZE	3	38:36
NS	1	39:39
ALLOWRETRY	1	40:40
ORDER	2	42:41
PCRDTYPE	4	46:43
LPID	5	51:47
GroupIDExt	3	54:52
EXPCOMPACK	1	55:55
RSVDC	8	63:56

The following table shows REQ channel width and bit ranges for the secondary match group.

Table 6-2: REQ channel: secondary match group

Field	Width	Bit range
QOS	4	3:0
ADDR	52	55:4

Field	Width	Bit range
LIKELYSHARED	1	56:56
MEMATTR	4	60:57
SNPATTR	1	61:61
EXCL/SNOOPME	1	62:62
TRACETAG	8	63:63

The following table shows REQ channel width and bit ranges for the tertiary match group.

Table 6-3: REQ channel: tertiary match group

Field	Width	Bit range
SRCID/TGTID	11	10:0
OPCODE	7	17:11
MPAM	11	28:18
TagOp	2	30:29
ADDR[38:6]	33	63:31

The following table shows RSP channel width and bit ranges for the primary match group.

Table 6-4: RSP channel: primary match group

Field	Width	Bit range
QOS	4	3:0
SRCID/TGTID	11	14:4
OPCODE	5	19:15
RESPERR	2	21:20
RESP	3	24:22
FWDSTATE/DATAPULL	3	27:25
CBUSY	3	30:28
DBID	12	42:31
PCRDTYPE	4	46:43
TagOp	2	48:47
TRACETAG	1	49:49
DEVEVENT	2	51:50
Reserved	1	52:52
Reserved	1	53:53

The following table shows SNP channel width and bit ranges for the primary match group.

Table 6-5: SNP channel: primary match group

Field	Width	Bit range
SRCID	11	10:0
FWDTXNID/{2'b0, STASHLPIDVALID, STASHLPID[4:0]}/VMIDEXT[7:0]	8	18:11

Field	Width	Bit range
FWDNID	11	29:19
OPCODE	5	34:30
NS	1	35:35
DONOTGOTOSD	1	36:36
RETTOSRC	1	37:37
TRACETAG	1	38:38
QoS	4	42:39
MPAM	11	53:43

The following table shows SNP channel width and bit ranges for the secondary match group.

Table 6-6: SNP channel: secondary match group

Field	Width	Bit range
SRCID	11	10:0
ADDR	49	59:11

The following table shows DAT channel width and bit ranges for the primary match group.

Table 6-7: DAT channel: primary match group

Field	Width	Bit range
QOS	4	3:0
SRC/TGTID	11	14:4
HOMENID	11	25:15
OPCODE	4	29:26
RESPERR	2	31:30
RESP	3	34:32
DATASRC/FWDSTATE/STASH	4	38:35
CBUSY	3	41:39
DBID	12	53:42
CCID	2	55:54
DATAID	2	57:56
POISON	4	61:58
DEVEVENT	2	63:62

The following table shows DAT channel width and bit ranges for the secondary match group.

Table 6-8: DAT channel: secondary match group

Field	Width	Bit range
SRC/TGTID	11	10:0
OPCODE	4	14:11
RESPERR	2	16:15

Field	Width	Bit range
RESP	3	19:17
TagOp	2	21:20
Tag	8	29:22
TU	2	31:30
DBID	12	43:32
TRACETAG	1	44:44
CHUNKV	2	46:45
DEVEVENT	2	48:47
RSVDC	8	56:49

6.1.2 DTM FIFO buffer

Traces captured by the DTM are stored in a four-entry DTM FIFO buffer. Each entry is 176-bits wide.

Entries are allocated to all enabled WPs as required. Trace data from WPs are packed based on the trace data format and stored within each entry to efficiently use the limited buffer storage. Therefore an entry might contain trace data from multiple flits captured at different times.

As each FIFO entry is filled, trace data from that entry is sent to the DTC for streaming out through the ATB interface.

6.1.2.1 Trace data format

CMN-700 supports several trace data formats.

The 3-bit packet type encoding in the DTM WP configuration register (por_dtm_wp{0..3}_config.wp_pkt_type) specifies the trace data format. The following table provides the supported trace data formats and their packet type encodings.

Table 6-9: Trace data formats

Packet type	Trace data format		Size	Max traces per FIFO entry
000	TXNID[11:0]		12 bits	14
001	{OPCODE[6:0],TXNID[11:0]}		19 bits	9
010	{3'b000,{TGTID[10:0], SRCID[10:0], OPCODE[6:0],TXNID[11:0]}		44 bits	4
011	Reserved		-	-
100	Control flit (see the following tables for field descriptions).	REQ	176 bits	1
		RSP	77 bits	
		SNP	119 bits	
		DAT	112 bits	
101	DATA[127:0]		-	-
110	DATA[255:128]		-	-

Packet type	Trace data format	Size	Max traces per FIFO entry
111	Reserved	-	-

Trace data is packed into a DTM FIFO buffer entry so that the higher-order bytes contain older trace data. For example, consider the following scenario:

- The trace data format is set to TXNID (type 0)
- Three TXNIDs (trace data) are received in the order `0x01`, followed by `0x02`, followed by `0x03`.

In this scenario, the trace FIFO entry is set to:

- `0000_0000_0000_0000_0000_0000_0000_0001_0203`

The following tables describe the control flit formats for various flit channels beginning with REQ control flit information, when MPAM is either enabled or disabled.

Table 6-10: REQ control flit

Field	Width	Bit range (MPAM enabled)	Bit range (MPAM disabled)
QoS	4	3:0	
TgtID	11	14:4	
SrcID	11	25:15	
TxnID	12	37:26	
StashTgtID / ReturnNID / SLCRepHint[6:0]	11	48:38	
StashTgtValid / Endian	1	49:49	
ReturnTID[9:0] / {`4'b0`, StashLPValid, StashLP[4:0]}	12	61:50	
Opcode	6	68:62	
Size	3	71:69	
NS	1	72:72	
LikelyShared	1	73:73	
AllowRetry	1	74:74	
Order	2	76:75	
PCrdType	4	80:77	
MemAttr	4	84:81	
SnpAttr	1	85:85	
LPID	5	90:86	
GroupIDExt	3	93:91	
Excl/SnoopMe	1	94:94	
ExpCompAck	1	95:95	
TagOp	2	97:96	
TraceTag	1	98:98	
MPAM	11	109:99	-
Addr	52	161:110	150:99

Field	Width	Bit range (MPAM enabled)	Bit range (MPAM disabled)
RSVDC	8	169:162	158:151
Total	170 (MPAM enabled) / 159 (MPAM disabled)	-	

The following table contains RSP control flit information.

Table 6-11: RSP control flit

Field	Width	Bit range
QoS	4	3:0
TgtID	11	14:4
SrcID	11	25:15
TxnID	12	37:26
Opcode	5	42:38
FwdState / `2'b0`, DataPull	3	45:43
RespErr	2	47:46
Resp	3	50:48
CBusy	3	53:51
DBID	12	65:54
PCrdType	4	69:66
TraceTag	1	70:70
DEVEVENT	2	72:71
Reserved	1	73:73
TagOp	2	75:74
Reserved	1	76:76
Total	77	-

The following table contains SNP control flit information, when MPAM is either enabled or disabled.

Table 6-12: SNP control flit

Field	Width	Bit range (MPAM enabled)	Bit range (MPAM disabled)
QoS	4	3:0	
SrcID	11	14:4	
TxnID	12	26:15	
FWDNID	11	37:27	
FWDTXNID[9:0] / {`4'b0`, StashLPValid, StashLP[4:0]} / {`2'b00`, VMIDExt[7:0]}	12	49:38	
Opcode	5	54:50	
NS	1	55:55	
DoNotGoToSD / DoNotDataPull	1	56:56	
RetToSrc	1	57:57	

Field	Width	Bit range (MPAM enabled)	Bit range (MPAM disabled)
TraceTag	1	58:58	
MPAM	11	69:59	-
Addr	49	118:70	108:60
Total	119 (MPAM enabled) / 108 (MPAM disabled)	-	

See also 7.11 DEVEVENT on page 1262 for more information.

The following table contains DAT control flit information.

Table 6-13: DAT control flit

Field	Width	Bit range
QoS	4	3:0
TgtID	11	14:4
SrcID	11	25:15
TxnID	12	37:26
HomeNID	11	48:38
Opcode	4	52:49
RespErr	2	54:53
Resp	3	57:55
FwdState / {2'b0, DataPull}	4	61:58
CBusy	3	64:62
DBID	12	76:65
CCID	2	78:77
DataID	2	80:79
TagOp	2	82:81
Tag	8	90:83
TU	4	94:81
TraceTag	1	95:95
RSVDC	8	103:96
Poison	4	107:104
CHUNKV	2	109:108
DEVEVENT	2	111:110
Total	112	-

See 7.11 DEVEVENT on page 1262 for more information.


Note

CHUNKV[1:0] denotes whether the upper or lower 128 bits of data are valid.

6.1.3 Read mode

Read mode provides an alternate way to access trace data that is stored in the DTM trace FIFO buffer, through configuration register access.

Each entry in the FIFO buffer is mapped to three 64-bit configuration registers, dtm_fifo_entry{0..3}_X, where X = 0, 1, or 2.

Read mode is enabled by setting the trace_no_atb bit in the DTM control register (por_dtm_control). Setting this bit causes all FIFO entries to be cleared and the DTM FIFO read status register (por_dtm_fifo_entry_ready) to be reset.

In this mode, each FIFO entry is allocated to the corresponding WP. For example, por_dtm_fifo_entry0_{0..2} is allocated to WP0 and por_dtm_fifo_entry1_{0..2} is allocated to WP1.

For packet types `0b000`, `0b001`, and `0b010`, only TXNID, OPCODE, SRCID, and TargetID are accumulated in the FIFO. The oldest transactions are shifted left to the MSB side of the FIFO, por_dtm_fifo_entry<0..3>_2. The newest transactions are on LSB side of the FIFO, por_dtm_fifo_entry<0..3>_0.


Note

When you program any DTM in read mode, ATB mode is not on so the ATB protocols, for example flush, do not function properly.

The read status for each WP trace data is reflected in the corresponding bit in the DTM FIFO entry ready status register (por_dtm_fifo_entry_ready). When a FIFO entry is full, the corresponding status bit is set, indicating that the trace data is ready to be read. Subsequent writes into that FIFO entry are disabled until the status bit is cleared. A write with a value of 1 clears the status bit and enables the corresponding FIFO entry to capture subsequent trace data.

6.1.4 DTC

DTCs control DTMs.

The main features of the DTC are:

- Trace packing, generation, and streaming through ATB interface
- Time stamping of traces
- Global synchronized cycle counters in all units, 16-bit
- ATB flush of DTM and DTC. AFREADY might be asserted several cycles after trace data output as DTC must receive all flush responses from DTMs.
- Watchpoint trigger event-based interrupt
- Eight sets of performance counters, 32-bit, with shadow registers, which are paired with one or more DTM local counters.

- PMU snapshot of DTM and DTC
- PMU overflow interrupt

See 4.4.8.2 Program DTC on page 1154 for the DTC programming sequence.


Note

When CMN is generating trace, the SOC has to do ATB flush before clock and power state transition

6.1.5 ATB packets

Each DTC has an ATB interface and generates ATB packets to send downstream through this interface. There are different varieties of ATB packets which are used for different functions.

Each DTC aggregates the flit trace data from the DTMs into the DTC trace FIFO, which packetizes them, and sends them out on its ATB interface. The DTCs also send other control and debug packets through this interface. There are various packet formats that are used on the ATB interface, which are described in the following sections:

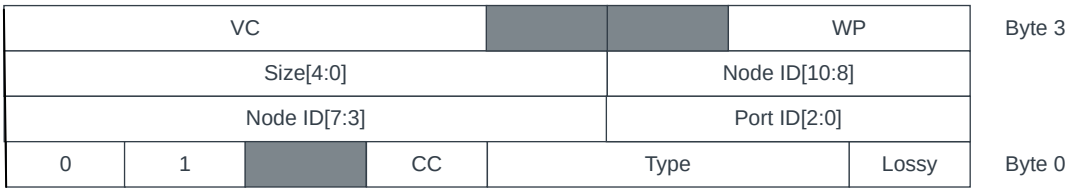
- 6.1.5.1 Trace data packet format on page 1212
- 6.1.5.2 Alignment sync packet format on page 1213
- 6.1.5.3 Time stamp packet format on page 1213
- 6.1.5.4 Cycle counting packet format on page 1214
- 6.1.5.5 Trace stream example on page 1215

6.1.5.1 Trace data packet format

Trace data packet contains a 4B header and a payload of variable size.

The following figure shows the packet header.

Figure 6-3: Trace data packet header



The packet header contains the following fields:

VC[3:2] CHI channel number for replicated channel.
VC[1:0] CHI channel.

00	REQ
01	RSP
10	SNP
11	DAT

PORT ID[2:0] Device port number
WP# Watchpoint number that captured the trace (0-3).
Type Packet format type.
Size Payload size, which is specified as (number of bytes – 1).
NodeID CHI node ID
CC Cycle counter. When set, this field indicates that a 2B cycle count is included in the packet after the payload.

The following key points must be observed:

- For packet type `100`, `101`, and `110` the leading zeros in upper-order bytes of the trace data payload are compressed and not transmitted. The payload Size field in the trace packet header is adjusted accordingly. For example, trace data = `0000_0000_0000_0000_0000_0000_0000_0001_0203` is sent as `01_0203`, with Size = 2 (indicating 3B transferred).
- The WP field selection for match might be different from the type of payload that is generated from the matching. When WPs are combined, the lower watchpoint number is specified as the WP# in the trace packet header.
- Trace data is of variable length. The expected number of bytes, not including the header, is (Size + 1). And with CC, another 2B are included at the end of the trace data.
- Anytime the previous packets cannot be transmitted in full, a lossy bit is asserted for the immediate next packet. A separate lossy bit is maintained for each of the watchpoints.

6.1.5.2 Alignment sync packet format

The alignment sync delimits trace start.

The alignment sync packet is 20B long and comprises 19B of zeros followed by `0x80`.

The alignment sync packet is the first packet that is sent after tracing is enabled. Also, you can configure the DTC to send the alignment sync packet periodically by programming the por_dt_trace_control register.

6.1.5.3 Time stamp packet format

The time stamp packet carries the SoC timer information. The time stamp is used to align the sequence of trace events across the SoC. To limit the number of bytes required to issue the timestamp, leading zero bytes of the timestamp value are suppressed.

On the first transmission of the timestamp, the timestamp value will be sent out in full. After that the timestamp is reconstructed by taking the previous timestamp value and replacing the lower order bytes with the new bytes that are sent in the new timestamp packet.

The time stamp packet is sent opportunistically under the following circumstances when the DTC FIFO has enough space to accommodate the time stamp packet:

- After each alignment sync packet is sent
- Periodically, based on the setting of the timestamp_period field of the por_dt_trace_control register.

The following figure shows the time stamp packet format.

Figure 6-4: Time stamp packet

The time stamp packet contains the following fields:

TS# 3-bit encoding of the size of time stamp that is specified as (number of bytes – 1).

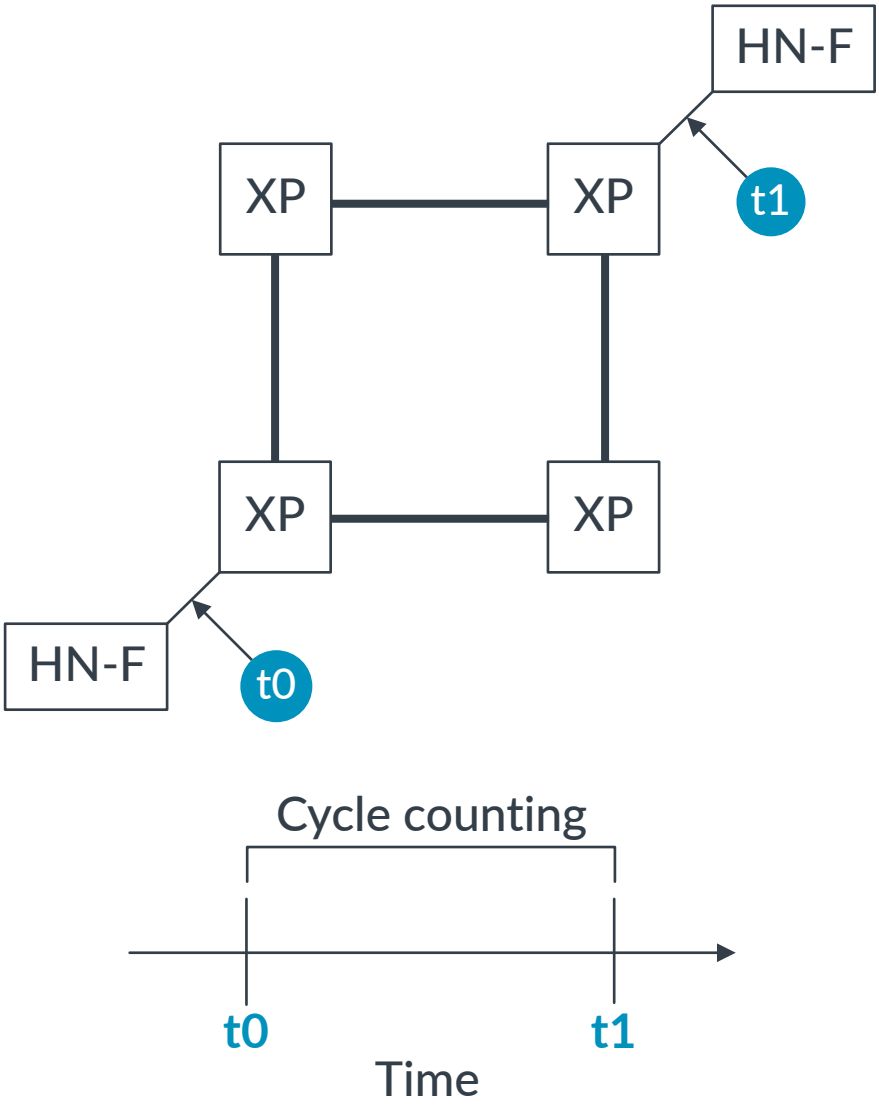
CC When set, indicates that a 2B cycle count is included in the packet after the payload.

6.1.5.4 Cycle counting packet format

Trace packets include an optional attached cycle counter.

Each watchpoint includes a configuration bit. The logical operator AND is used on the configuration bit and global cycle count enable. The following figure shows a typical cycle counting scenario.

Figure 6-5: Cycle counting



The cycle counter payload is 2B. The CC bit in the trace packet header indicates the cycle counter payload. Cycle counters across the interconnect are turned on and off synchronously. This feature ensures that all of them have the same time stamp value.

6.1.5.5 Trace stream example

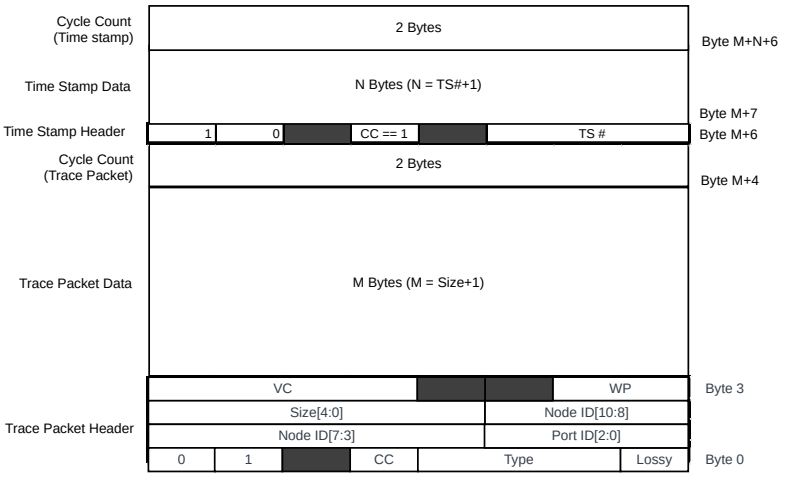
DTCs send out trace data on the ATB bus as a trace stream.

The following figure shows an example trace stream. It consists of:

- 4B trace packet header
- <*M*>B trace data
- 2B cycle count

- 1B time stamp header
- <*N*>B time stamp
- 2B cycle count

Figure 6-6: Trace stream



6.2 DT usage examples

To help you use the CMN-700 DT features, we describe some example use cases of the DT system and example programming for those use cases.

See the following sections:

- 6.2.1 Flit tracing on page 1216
- 6.2.2 Trace tag on page 1218
- 6.2.3 Debug watch trigger events on page 1221
- 6.2.4 Cross trigger on page 1221

6.2.1 Flit tracing

CMN-700 can trace individual flits at device interfaces at each XP.

You can program DTM WPs to monitor flit uploads and downloads at each of the two XP device ports on any of the four CHI channels:

- REQ
- RSP
- SNP

- DAT

You can use a set of value and mask registers to define a subset of flit fields that are then used for matching at the WP.

On a match, WPs capture and store flit fields into trace buffers so that they can be used for debug. The generated trace can then be streamed out on the ATB interface or accessed using a control register interface.

For more information about the format of the value and mask registers, and the format of trace packets, see 6.1.1.1 WP match value and mask register on page 1203 and 6.1.2.1 Trace data format on page 1207.

6.2.1.1 Flit tracing example

CMN-700 can trace individual flits at device interfaces at each XP.

See 4.4.8.1 Program DTM watchpoint on page 1153.

The following shows an example of setting up a simple trace of REQ flits. The example corresponds to a ReadShared transaction to address = X initiated by RN-F2 and sending out the trace packets on the ATB bus.

To monitor REQ flits uploaded from RN-F2, set up watchpoints (WPs) inside XP connected to RN-F2. The Opcode and Addr fields are mapped to the primary and secondary match registers respectively. Therefore, you must set up two WPs, one to monitor the Opcode and the other to monitor the Addr.

To set up these WPs:

1. Program WP0 (upload WP) to monitor REQ.Opcode:
    a. Set dtm_wp0_val/mask registers to match on Opcode = ReadShared
    b. Set dtm_wp0_config to:
        1. Select upload device port ({wp_dev_sel2, wp_dev_sel} = RN-F2_port)
        2. Select flit channel (wp_chn_sel = REQ)
        3. Match format group to primary for Opcode match (wp_grp = 0)
        4. Set combined mode to gang-up WP0 and WP1 (wp_combine = 1)
        5. Enable REQ flit trace packet generation (set wp_pkt_type and wp_pkt_gen = 1)
2. Program WP1 (upload WP) to monitor REQ.Addr as follows:
    a. Set dtm_wp1_val/mask registers to match on Addr = X.
    b. Set dtm_wp1_config to:
        1. Select upload device port ({wp_dev_sel2, wp_dev_sel} = RN-F2_port)
        2. Select flit channel (wp_chn_sel = REQ)
        3. Match format group to secondary for Addr match (wp_grp = 1)

Combined mode and trace generation

In combined mode, WP0 config settings are used to enable trace generation.

To enable trace generation in the WP, program WP0 as follows:

1. Set dtm_control.dtm_enable = 1 and program the DTC to start the trace
2. Program por_dt_traceid.traceid according to the *Arm® CoreSight™ Architecture Specification*. The supported range of values is `0x01-0x6F`.
3. Program por_dt_dtc_ctl to enable tracing (dt_en = 1)

6.2.2 Trace tag

CMN-700 can generate trace tags at the device interfaces and propagate them to destination devices.

This feature enables a set of flits corresponding to a specific transaction or set of transactions that match a specific criterion to be tagged for tracing.

For example, using the trace tag mechanism, flits from all four CHI channels can be monitored:

- REQ
- RSP
- SNP
- DAT

An example of a monitored transaction is a memory read transaction to a specific address, which is then tagged for tracing.

6.2.2.1 Trace tag generation

Internal XPs and external RN-F or SN-F devices can generate a trace tag. The generated trace tag is reflected in the TRACETAG field of the uploaded flit.

Inside the XP, DTM WPs generate the trace tag by matching on flits uploaded at the corresponding XP device port. The DTM WPs can be programmed to match on a flit on any of the four CHI channels: REQ, RSP, DAT, and SNP.

The WP generates a trace tag when there is a match and the trace_tag_enable field of the por_dtm_control register is set to 1.

You can program DTM WPs to match on a flit on any CHI channel and on any device port. However, for debug, it is most useful to program WPs to match on REQ flits uploaded at the RN

and HN-F device ports. We recommend this programming because REQ flits are the starting flits that originate new transactions. When tagged, subsequent RSP, SNP, and DAT flits that relate to the same transaction carry the trace tag.

If the following conditions are all true, the XP does not generate the trace tag:

- Flit transfer takes place from one device port to the other device port, within the same XP.
- The flit destination device is not an HN.
- The flit transfer to its destination occurs one cycle after the XP receives it.

6.2.2.2 Trace tag propagation

All CMN-700 devices forward on the trace tag, when asserted, from a received flit corresponding to a transaction to all subsequent flits associated with that transaction.

The HN-F also propagates the trace tag from the source transaction to spawned transactions such as SLC evictions and SF back invalidations.

Using the logical operator OR, the trace tag that is generated inside the XP is combined with the TRACETAG field of the received flit. The resultant value is then sent in the TRACETAG field of the flit transmitted to the destination device.

6.2.2.3 Trace tag trace packet generation

If a WP is set up to generate trace packets, then flits with trace tags can trigger trace packet generation. For this process to occur, the flit must also be on the CHI channel and device port that the WP is monitoring..

The wp_pkt_gen field of the por_dtm_wpN_config registers enables trace packet generation for a specific WP. If trace packet generation is enabled, then any flit that the WP sees with the TRACETAG field asserted generates a trace packet. The selected CHI channel and device port determine which flits the WP sees.

The following fields of the por_dtm_wpN_config register determine the CHI channel and device port that the WP monitors:

- The wp_chn_sel field determines the WP CHI channel.
- The wp_dev_sel2 and wp_dev_sel fields determine the WP device port.

If a trace packet is generated, the wp_pkt_type field of the por_dtm_wpN_config register determines the type of trace packet that the WP generates.

This trace packet is generated whenever TRACETAG is asserted in a flit. A WP match on the por_dtm_wpN_val and por_dtm_wpN_mask register values is not required.

6.2.2.4 Trace tag example programming

This example programming outlines a trace tag scenario-based trace generation with synchronized cycle counts.

For more information about watchpoint programming, see 4.4.8.1 Program DTM watchpoint on page 1153.

This example programming sets up a simple trace of REQ flits. The example corresponds to a ReadShared transaction to address=X. RN-F 2 initiates the transaction in the mesh and the WP sends trace packets out on the ATB bus.

To monitor REQ flits uploaded from RN-F 2, set up WPs inside the XP that RN-F 2 is connected to. For REQ flits, the Opcode and Addr fields are mapped to the primary and secondary match registers respectively. Therefore, you must set up two WPs, one to monitor the Opcode and the other to monitor the Addr.

To set up these WPs:

1. Program WP0 (upload WP) to monitor REQ.Opcode:
   a. Set por_dtm_wp0_val and por_dtm_wp0_mask registers to match on Opcode=ReadShared
   b. Set por_dtm_wp0_config to:
      1. Select upload device port ({wp_dev_sel2,wp_dev_sel} = RN-F 2 port)
      2. Select flit channel (wp_chn_sel = REQ)
      3. Match format group to primary for Opcode match (wp_grp = 0)
      4. Set combined mode to gang-up WP0 and WP1 (wp_combine = 1)
      5. Enable REQ flit trace packet generation (set wp_pkt_type and wp_pkt_gen = 1)
2. Program WP1 (upload WP) to monitor REQ.Address:
   a. Set por_dtm_wp1_val and por_dtm_wp1_mask registers to match on Addr = X
   b. Set por_dtm_wp1_config to:
      1. Select upload device port ({wp_dev_sel2, wp_dev_sel} = RN-F 2 port)
      2. Select flit channel (wp_chn_sel = REQ)
      3. Match format group to secondary for Addr match (wp_grp = 1)

In combined mode, the configuration settings for WP0 are used to enable trace generation, using the following steps:

1. Enable trace tag generation in the WP:
   - Set the trace_tag_enable field of the por_dtm_control register = 1
   - Set the dtm_enable field of the por_dtm_control register = 1 and program the DTC to start the trace
2. Program the trace id field of the por_dt_traceid register according to the *Arm® CoreSight™ Architecture Specification*. The supported range of values is `0x01-0x6F`.

3. Program por_dt_dtc_ctl to enable tracing (dt_en = 1)

6.2.3 Debug watch trigger events

DTM WPs can be programmed to match on specific flits and generate a debug watch trigger event to the DTC.

You can program the DTC to signal the debug watch trigger event in one or both of the following ways:

- Signal a debug watch trigger interrupt on the DBGWATCHTRIGREQ/DBGWATCHTRIGACK interface.

  
  Note

  This interface is based on a four-phase handshake protocol.

- Signal an ATB trace trigger with ATID `0x7D` on the ATB interface.

In a system with multiple DTCs, each DTC has its own ATB interface on which it signals ATB trace triggers from DTMs within its DTC domain. Multiple DTCs also have their own DBGWATCHTRIGREQ / DBGWATCHTRIGACK interfaces on which they signal debug watch trace interrupts.

6.2.4 Cross trigger

CMN-700 can trigger DTMs based on specific events occurring elsewhere in the system.

By default, DTMs start monitoring and tracing flits without waiting for another event. The cross trigger feature allows flit monitoring and tracing to be delayed until after the events of interest are observed in the system.

The cross trigger event is set up in two steps:

1. Set up DTM WPs to monitor flits and generate traces.
2. Other DTM WPs (in the same or different XPs) are set up to generate a cross trigger on specific events to the DTC which triggers the DTMs in step 1.

6.2.4.1 Cross trigger example

CMN-700 can trigger DTMs based on specific events occurring elsewhere in the system.

See 4.4.8.1 Program DTM watchpoint on page 1153.

This example uses trace DAT flits corresponding to a ReadShared transaction to address-X that originated at RN-F2. There have also been 10 WriteNoSnoops uploaded to the HN-D.

Set WP or WPs at all DAT download ports to generate DAT flit traces for ReadShare transactions from RN-F2 to address-X:

1. Program WP2 and WP3, which are at the DAT download ports, to trace DAT flits:
   a. Set dtm_wp2_val/mask and dtm_wp3_val/mask registers to match on SRCID = RN-F2, opcode = ReadShared, and address = X
   b. Set dtm_wp2_config and dtm_wp3_config to the respective download device ports (wp_dev_sel = 0 and wp_dev_sel = 1). Select the DAT channel by setting the wp_chn_sel bit to the DAT encoding.
2. Enable the WP by setting the dtm_enable bit of the dtm_control register to 1

Set up WP at HN-D upload port to monitor WriteNoSnoop flits. Program WP0 (upload WP) to monitor and enable cross trigger REQ. Opcode as follows:

1. Set dtm_wp0_val/mask registers to match on Opcode = WriteNoSnoop.
2. Set dtm_wp0_config to:
   a. Select upload device port (wp_dev_sel = HND_port)
   b. Flit channel (wp_chn_sel = REQ)
   c. Match format group to primary for Opcode match (wp_grp = 0)
   d. Enable cross trigger (wp_ctrig_en = 1)
3. Enable WP: Set dtm_control.dtm_enable = 1

Set up counter in DTC to count ten trigger events from step 3. Program por_dt_dtc_ctl as follows:

1. Set cross trigger count (cross_trigger_count = 10)
2. Enable waiting for HN-D WP trigger event (dt_wait_for_trigger = 1)
3. Enable DTC (dt_en = 1)

6.2.4.2 Sample profile

CMN-700 supports the Armv8.2 sample extension.

WPs can be programmed to monitor channel, opcode, and related PM items. The sampling interval counter register, por_dtm_pmsicr, counts down with each match. When the counter reaches zero, the trace tag field of the next matched transaction is asserted. At the same time, the counter is reloaded with the programmed value from the sampling interval reload register, por_dtm_pmsirr, and the next countdown cycle begins.

If the CMN-700 configuration uses a single DTM per XP, then only one outstanding transaction is expected. Therefore, there is only one set of por_dtm_pmsicr and por_dtm_pmsirr registers per XP. Por_dtm_pmsicr is 24 bits, and the lower 8 bits of por_dtm_pmsirr are zero.

Note

If your configuration uses multiple DTMs per XP, then extra sample profile registers are instantiated.

Secure transactions can be tagged and traced when secure debug is enabled, see 6.4 Secure debug support on page 1224 for more details.

6.3 Performance Monitoring Unit system overview

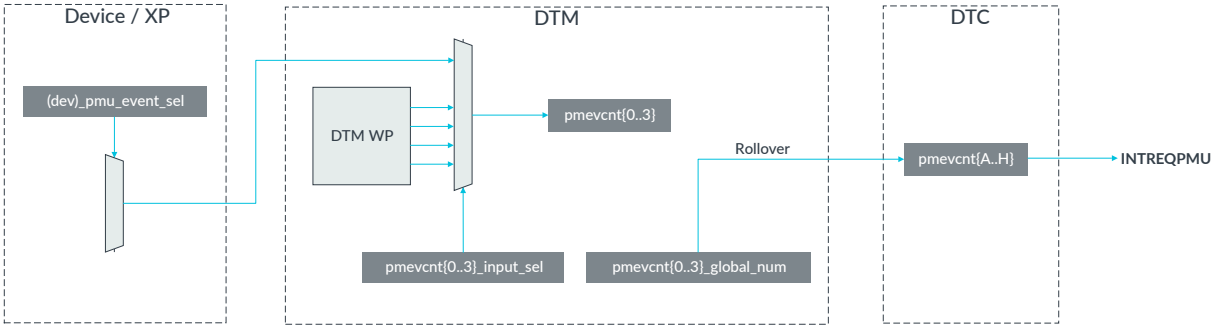
CMN-700 includes *Performance Monitoring Unit* (PMU) capabilities.

The PMU offers the following features:

- Local and global performance counters with shadow registers
- PMU snapshot across all internal CMN-700 devices

The PMU consists of local performance counters in the DTMs and global performance counters in the DTCs. The following figure shows this structure of local and global performance counters.

Figure 6-7: PMU local and global performance counters



For the various PMU programming sequences, see 4.4.9 PMU system programming on page 1154.

The PMU system performs the following tasks:

- Selects PMU event from XP, the local watchpoint, and the devices on XP device ports
- Operates four local PMU counters, 4 × 16b
- Operates eight global PMU counters, 8 × 32b, associated with the local counters.
- Snapshot.
- Overflow interrupt from global PMU counters.

The PMU counter value can be copied over into the shadow registers when there is:

- A request of snapshot through input pin PMUSNAPSHOTREQ

- A write into the ss_req field of the 4.3.7.23 por_dt_pmsrr on page 538 register within the DTC

On receiving a snapshot request, a DTC sends a snapshot request to all DTMs. Multiple snapshot requests are collapsed into a single request. On receiving PMU snapshot packets from all DTMs, rollover information is updated at the global counters, and the counter value is copied to the shadow registers.

6.4 Secure debug support

The SPNIDEN input and the value of the secure_debug_disable field of the por_dt_secure_access register control the Secure debug state.

Secure debug is enabled when SPNIDEN is asserted, or when the secure_debug_disable bit of the por_dt_secure_access register is set to 0. The default value of this bit is 0.

When Secure debug is enabled, all events can be counted and all flits can be traced.

When Secure debug is disabled, all events with **UNKNOWN** Secure state are not counted and all flits with **UNKNOWN** Secure state are not traced.

7. Performance optimization and monitoring

This chapter describes the performance optimization techniques and *Performance Monitoring Unit* (PMU) that System integrators can use to optimize the functionality of the interconnect implementation.

7.1 Performance optimization guidelines

There are some restrictions when optimizing CMN-700.

To obtain maximum performance from CMN-700, the system integrator must be aware of the following information:

RN-I

When request ordering is not required, for best performance transaction requests must be dispatched with non-overlapping IDs to ensure optimal bandwidth operation. Large Burst transactions, that are larger than 64B, must be split into 64B or smaller Burst transactions for best performance. In addition, set AxSIZE to the AXI bus width of the RN-I to fully utilize the available bandwidth.

For example, if the AXI bus width is:

128b Set AxSIZE = 4 (16B).
256b Set AxSIZE = 5 (32B).
512b Set AxSIZE = 6 (64B).

Read or write requests to different parts of the same cache line must be combined into a single cache line request for best performance. 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.

Based on the transaction attributes, RN-I can enforce more ordering on transactions, targeting device memory downstream of HN-I, affecting the overall achievable bandwidth.

RN-I and RN-D also support disabling of read data interleaving using the s*_dis_data_interleaving configuration bits in the por_rnd_s_0-2_port_control registers. This setting only applies to data returned on the RDATA channel, if otherprops is g.signal.name, in response to read requests from AR channels. Disabling of read data interleaving does not apply to atomic operations. For example, the RDATA data, if otherprops is g.signal.name, data from an atomic operation sent on the AW channel can interleave with RDATA data, if otherprops is g.signal.name, data from an AR channel request.

When any one port's s*_dis_data_interleaving is set, each AXI port's requests will be routed to a single RRT arslice. Each port will route to a unique arslice if the number of RRT arslices is 4 or 8. If there are only two arslices, port0 will go to arslice0, while port1 and port2 will both go to arslice1.

When s*dis_data_interleaving is set, all requests are treated as same-ARID, request will issue without dependency unless they are same-address. Responses will be returned in order, after all previous older request have completed.

When s*dis_data_interleaving is set, read bursts are always cracked. There is no read burst preservation possible.

Due to the above behaviors performance is expected to be diminished in this mode.

RN-I and RN-D also includes an alternative method for disabling data interleaving for better performance that can be used if certain system requirements are met. If these requirements are not met deadlocks are possible. The system must be well understood, if enabling this feature. The configuration bit for this alternative mode is sys_dis_data_interleaving and it is in the RN-X's aux_ctl register. It is required that this bit is set to 0 if any of the port s*_dis_data_interleaving are set, and likewise, that the port bits are all set to 0 when the sys_dis_data_interleaving bit is set.

The following are the requirements associated with SYS_DIS_DATA_INTERLEAVING aux_ctl bit in RN-X:

1. Cannot set SYS_DIS_DATA_INTERLEAVING for multi-chip systems. Support for remote HN-P is a future feature.
2. When setting SYS_DIS_DATA_INTERLEAVING for an RN-I/RN-D it is required for that RN-I/RN-D to target only one HN-P with read bursts. It is also required for the path to a single remote HN-P, when remote support is available, to only go through a single CCG and for the ccg_rni to also have SYS_DIS_DATA_INTERLEAVING set.
3. The AXI subordinate downstream of HN-P must not interleave read burst data.
4. Number of entries requirement
    a. Must have NUM_RD_REQ_PARAM == NUM_RD_BUF_PARAM (disabled for decoupled mode, setting SYS_DIS_DATA_INTERLEAVING will have no effect)
    b. Must have NUM_RD_REQ_PARAM<=256 for timing reasons (disabled for RRT512, setting SYS_DIS_DATA_INTERLEAVING will have no effect)
5. The sum of the maximum number of beats of a cracked burst per each port must fit wholly within an arslice - breaking this rule will result in a hang.
    a. To simplify, in single port use case, ARLEN can be restricted to ARLEN<RNID_NUM_XRT_SLICE_REQ
        1. For example, when RNID_NUM_XRT_SLICE_REQ is 64, (4 arslices of 64 entries, 256 total entries). ARLEN should then be restricted to a max value of 63.
        2. Otherwise need to comprehend narrow bursts, axi mem_attr, cracking and packing behavior, to ensure cracked bursts fit in completely within the arslice.
    b. As an example, in multiple port use case, if max cracked beats allowed from port0 is 32 and from port1 is also 32, and port2 is not used. This is allowed because 32+32 would fit within a 64 entry slice. However, if max cracked beats from each port0 & port1 is 64, then they

couldn't both fit concurrently in the arslice. Note the details below that affect availability of arslice entries.

c. Recommend to keep DIS_PORT_TOKEN=1 (wouldn't be useful in the single port use case). See 3.1.4.4 Dedicated RN-I resources for AXI port traffic on page 73 for description of DIS_PORT_TOKEN functionality.

d. Recommend to keep QPC15_ENTRY_RSV_EN=0, which would otherwise reserve one entry for QoS15 requests.

e. Not following c&d could prevent incoming requests from wholly fitting in an arslice due to reserved entries limiting the number of entries available for a cracked burst to use preventing the whole cracked burst from fitting within the arslice.

f. For TCU traffic on a second port of RN-I/RN-D, DIS_PORT_TOKEN must be set to 0 to guarantee forward progress of the TCU. It is recommended for por_rni_s#{index}_port_control.s#{index}_rd_token to also be set to zero so that each port only reserves one entry. The impact of this on the number of free arslice entries must be comprehended in determining the max cracked burst size supported. The sum of the maximum number of beats of a cracked burst for each port must then be less than the arslices' number of entries-2 (where 2 represents 1 reserved entry for each of the other two ports) in this case.

Description of the new system disable data interleaving behavior:

1. No data interleaving. Atomics also do not interleave
2. Bursts are now preserved to HNP using normal burst preservation requirements listed, see PCIe read Burst preservation through the interconnect .
3. All arslices are utilized (axid hashed, same as in normal mode)
4. All requests can still re-order
5. Cracked requests wait for all burst data for all associated cracked beats to be returned on CHI before arbitrating out to AXI RChannel
6. Entry and Slice arbiter are sticky on RLAST to prevent data interleaving.
7. SameID can be used for cracked bursts
8. Multiple AXI read ports can be used following precautions of 5 in the requirements list above. Use caution when using this mode.

RN-I and RN-D upsizes a subcacheline ReadOnce transaction arriving on AXI to a 64B ReadOnce request on CHI.

This behavior can result in extra data beats on the interconnect. To prevent data overhead, RN-I and RN-D added a feature to change opcode to ReadNoSnoop on CHI while keeping original transaction size. Note that this feature relies on the invisible cache behavior of HN-F. By default, this feature is not enabled if target type is CML RA. If all CCG RAs in the system are connected to an SMP CML link, then there are specific programming requirements to enable this feature in CML RA. In this case, the en_subcacheline_rdonce_conv bit of the por_rni_cfg_ctl register must be programmed.

RN-I and RN-D support preservation of AXI Burst for PCIe read transactions through the interconnect when the request is targeting HN-P and PCI_CXRA node types, see PCIe read Burst preservation through the interconnect.

RN-I and RN-D also support AXID-based targetID selection for HN-P, CML RA, PCI_CXRA targets (see RN-I and RN-D AxID -based target selection).

HN-F

High temporal locality of address usage in transactions can cause same-address dependencies to occur for transactions with addresses to overlapping cache lines. This condition results in higher latency because of serialization delays between these transactions. CMN-700 is microarchitected to avoid hot spotting in the HN-F partitions or in the memory controllers. However, this condition is unavoidable in cases of temporally local same-address usage.

HN-I

Stream of interleaved reads and writes targeting the same peripheral downstream of HN-I results in higher latencies on these transactions. It also could result in serialization delays between these reads and writes. Arm recommends that read and write transactions are not interleaved when targeting the same peripheral.

7.1.1 RN-I and RN-D write Burst cracking

RN-I and RN-D crack write Bursts into individual CHI transactions.

In HN-P, AWUSER[MSB] is used to indicate the last transaction of a Burst from RN-I or RN-D. The last indication and AWID can be used to gather write transactions from a Burst downstream of HN-P.

Device memory write transactions (AWCACHE[1] == 0) must not be gathered downstream of HN-P.

For information on the AWID used for external write gathering logic, see B.2.4 HN-I and HN-P AxID signal properties and encodings on page 1281

7.1.2 RN-I and RN-D write data cancel

Write Data Cancel is a feature of the RN-I and RN-D, that includes a counter per write request entry.

The counter begins counting at the time a DBID response is received, and it counts so long as the entry has not become head of the data dependency chain. The counter terminates when a request entry becomes head of the data dependency chain, and can issue data. Or when the counter expires causing the issue of a write data cancel, instead.

When the counter expires causing the issue of a write data cancel, all younger writes on the same dependency chain must also be cancelled. Then all cancelled writes are re-issued in order, but in serialized fashion waiting on completions to arrive before re-issuing the next write request. New write requests arriving after a write data cancel has occurred are not serialized and return to normal write request behavior.

A write request entry uses the local timeout value if it holds a request to a local target, otherwise it uses the remote timeout value. For more information about the total approximate cycles of the local and remote timeout values before expiring, see 4.3.14.6 por_rnd_cfg_ctl on page 971 and 4.3.15.6 por_rni_cfg_ctl on page 994

7.1.3 PCIe read Burst preservation through the interconnect

RN-I and RN-D can preserve AXI Bursts for PCIe read transactions to HN-P and PCI_CXRA through the interconnect under certain conditions.

If the following conditions are met, then RN-I and RN-D can preserve AXI Bursts for PCIe read requests to HN-P and PCI_CXRA:

- `NUM_RD_BUF` == `NUM_RD_REQ`, or `NUM_RD_BUF` < `NUM_RD_REQ` and AXI chunking is enabled
  - If target is `PCI_CXRA` and rnsam_clstr_intlv_en is set, additional requirement that total burst size ARSIZE*ARLEN must be less than or equal rnsam_clstr_mask.
- RN-I or RN-D `AXDATA_WIDTH` parameter value is 512 or 256
- HN-P AXDATA_WIDTH parameter value is 512 or 256
- ARCACHE[1] == 1
- ARBURST == INCR
- ARSIZE == the RN-I or RN-D `AXDATA_WIDTH` parameter
- Read transactions must be unique ID transactions. Unique ID transactions are detected using one of the following properties:
  - ARIDUNQ indication on the transaction
  - Setting the s#{index}_sends_ar_uniq_id bit of the por_{rni, rnd}_s_{0-2}_port_control register to 1 for the given port
- The pcie_mstr_present bit of the por_{rni, rnd}_cfg_ctl register must be set to 1
- The s#{index}_dis_data_interleaving bit of the por_{rni, rnd}_s_{0-2}_port_control register must be set to 0.


Note

The aux_ctl bit for sys_dis_data_interleaving can be set and will still allow bursts to be preserved

---

- When a request does not target HN-P and PCI_CXRA or does not follow the preceding restrictions, Bursts are cracked into:
  - 64B chunks for normal memory transactions

- ARSIZE chunks for device memory transactions

- The read Burst preservation feature is enabled only for normal memory transactions (ARCACHE[1] == 1). For device memory transactions, an AXI Burst is not preserved through the interconnect and is cracked into ARSIZE chunks.

When the HN-P `AXDATA_WIDTH` is not 512 or 256, por_{rni,rnd}_aux_ctl.dis_hnp_rd_burst must be set to 1.

7.1.4 RN-I and RN-D AxID-based target selection

The RN-I and RN-D RN SAM can be configured to enable AxID-based target ID selection for HN-P, CML RA, and PCI_CXRA targets.

This feature enables higher bandwidth by striping traffic across multiple targets. The target ID within a *Hashed Target Group* (HTG) in the RN SAM can be programmed in a different order for a different RN-I or RN-D for a given address range. This functionality avoids targeting the same target ID when two or more RN-Is or RN-Ds receive traffic with overlapping AxID values. See 3.4.6 RN SAM on page 112.

7.1.5 CMN buffer lifetime and recommended parameter settings

The following tables contain information on the recommended buffer parameter settings and buffer lifetime for devices supported by the CMN-700.

Table 7-1: Buffer parameter settings: RN-D

Buffer	Parameter	Usage	Typical Latency Covered	Recommended setting/Latency covered	Comments
Read Request Tracker	`NUM_RD_REQ`	Outstanding Read Requests	Full round trip read completion	256 / 256ns	
Read Data Buffer	`NUM_RD_BUF`	Outstanding Read Requests	Full round trip read completion*	256 / 256ns	When read bursts are preserved across RND, a read burst takes one Request Tracker entry and one Data Buffer entry. See 7.1.3 PCIe read Burst preservation through the interconnect on page 1229
Write Request Tracker	`NUM_WR_REQ`	Outstanding Write Requests	Full round trip write completion	128 / 128ns	Write bursts are always cracked, and each cacheline beat takes one write request tracker entry.

Buffer	Parameter	Usage	Typical Latency Covered	Recommended setting/Latency covered	Comments
Write Data Buffer	NUM_WR_REQ	Outstanding Write Requests	Full round trip write completion*	128 / 128ns	

Table 7-2: Buffer parameter settings: HN-I

Buffer	Parameter	Usage	Typical Latency Covered	Recommended setting/Latency covered	Comments
Request Receive Tracker (RRT)	NUM_RRT_REQS	Prepare request for AXI dispatch	Read: Allocation to Dispatch Write: Round trip DBID	32 / 32ns	-
Request Dispatch Tracker (RDT)	NUM_AXI_REQS	Outstanding AXI requests	Full round trip request completion	64 / 64ns	-

Table 7-3: Buffer parameter settings: HN-P

Slice	Buffer	Parameter	Usage	Typical Latency Covered	Recommended setting / Latency covered	Comments
NP2P	Request Receive Tracker (RRT)	NUM_RRT_REQS	Prepare request for AXI dispatch	Read: Allocation to Dispatch Write: Round trip DBID	32 / 32ns	Used for CPU and non-PCIe RNI/D traffic only
	Request Dispatch Tracker (RDT)	NUM_AXI_REQS	Outstanding AXI requests	Full round trip request completion	64 / 64ns	
P2P Write	Request Receive Tracker (RRT)	WR_NUM_RRT_REQS	Prepare request for AXI dispatch	Write: Round trip DBID	32 / 32ns	Write bursts are not preserved, so each 64B or smaller write from RN-I/D will take one RRT and WDB entry from CHI and one RDT entry upon dispatch to AXI.
	Request Dispatch Tracker (RDT)	WR_NUM_AXI_REQS	Oustanding AXI requests	Full round trip request completion	64 / 64ns	-

Slice	Buffer	Parameter	Usage	Typical Latency Covered	Recommended setting / Latency covered	Comments
P2P Read	Request Dispatch Tracker (RDT)	RD_NUM_AXI_REQS	Oustanding AXI requests	Full round trip request completion	64 / 64ns	Read bursts are supported for 256-bit or 512-bit data widths only. Each burst takes one RRT tracker entry upon receipt from CHI and one RDT tracker entry upon dispatch to AXI. Returning data is passed directly to CHI.

Table 7-4: Buffer parameter settings: CCG for CML and CXL

Buffer	Parameter	CML SMP		CXL Host		Recommended setting/ Latency covered	Comments
		Usage	Type Latency Covered	Usage	Typical Latency Covered		
RA Request Tracker	RA_NUM_REQS	Outastanding requests	Full round trip remote completion	CXL outstanding M2S Req and RwD requests	Round trip remote completion	256 / 256ns	
RA Snoop Credit Buffer	RA_NUM_SNPREQS	Snoop credits	Round trip CML/SMP transport latency	Not Applicable		128 / 128ns	
HA Snoop Tracker	HA_NUM_SNPREQS	Outstanding Snoops	Round trip remote snoop completion	CXL OT H2D Requests (snoops)	Round trip remoe snoop completion	256* / 256ns	depends on remote
HA request Credit Buffer	HA_REQ_PASS_BUFF_DEPTH	Request credits	Round trip CML/SMP transport latency	CXL D2H Request credits	Round trip CXL latency	128 / 128ns	
HA Data Credit Buffer	HA_DAT_PASS_BUFF_DEPTH	Data credits	Round trip CML/SMP transport latency	CXL D2H Data credits	Round trip CXL latency	128 / 128ns	
LA Data Buffer	CCLA_RX_STL_BUFFER_DEPTH	Link credits	Round trip CML/SMP transport latency	CXL S2M Data credits (S2M DRS)	Round trip CXL latency	128 / 128ns	

Table 7-5: Buffer parameter settings: CCG for CHI

Buffer	Parameter	CHI	Typical Latency Covered	Recommended setting	Comments
HA Request Tracker	HA_NUM_REQS	Number of outstanding CHI Requests	Round trip local completion	192*	128 Reads + 64 Writes
HA Write Data Buffer (WDB)	HA_NUM_WRBUF	Number of outstanding CHI write data	Round trip DBID latency to local HN	64	
HA Read Data Buffer (RDB)	HA_NUM_RDBUF	Number of outstanding CHI read requests	Round trip local read completion	128	
HA Snoop Data Buffer	HA_NUM_SNPBUF	-	*Used to withstand any mesh stalls	32*	To withstand any mesh stalls. 32 is recommended for larger configs
RA Snoop Tracker	RA_NUM_SNPBUF	Number of active CHI snoops	Round trip local completion	32	
RA Read Data Buffer	RA_NUM_RDBUF	-	*Used to withstand any mesh stalls	32*	To withstand any mesh stalls. 32 is recommended for larger configs
RA Write Data Buffer	RA_NUM_WRBUF	Number of outstanding DBIDs	Round trip DBID latency to local RN	32	

Table 7-6: Buffer parameter settings: CCG_RNI

Buffer	Parameter	Usage	Typical Latency covered	Recommended setting / Latency covered	Comments
Read Request Tracker	NUM_RD_REQ	Outstanding Read Requests	Full round trip read completion	256 / 256ns	
Read Data Buffer	NUM_RD_BUF	Outstanding Read Requests	Full round trip read completion*	256 / 256ns	When read bursts are preserved across RND, a read burst takes one Request Tracker entry and one Data Buffer entry. See Section 6.1.2 PCIe read Burst preservation through the interconnect.
Write Request Tracker	NUM_WR_REQ	Outstanding Write Requests	Full round trip write completion	128 / 128ns	Write bursts are always cracked, and each cacheline beat takes one write request tracker entry.
Write Data Buffer	NUM_WR_REQ	Outstanding Write Requests	Full round trip write completion*	128 / 128ns	

Table 7-7: Buffer parameter settings: SBSX

Buffer	Parameter	Lifetime	Typical Latency covered	Recommended setting / Latency covered
DART	SBSX_NUM_DART_PARAM	Request to Response	Full roundtrip read completion	128 / 128ns
Write Data Buffer	SBSX_NUM_WR_BUF_PARAM	Request to last beat of datasent.	Full roundtrip write completion	16 / 16ns

7.2 About the Performance Monitoring Unit

CMN-700 provides access to various performance events. Some of these events are unique to and originate in a specific CMN-700 component. Some of the events are available by using watchpoints in the DTM watchpoint in the XP where the component is located.

The PMU input source must be configured to select the watchpoint input, according to the instructions in 4.4.9.1 Set up PMU counters on page 1154.

This chapter describes the performance events and the relevant use cases for most of those events. For information about the infrastructure and logic that enable general utility of the performance monitor events, see 6. Debug trace and PMU on page 1200.

7.2.1 Cycle counter

The cycle counter is used to track time.

You can reset this counter to initiate the time interval over which you want to capture the events.

PMU_CYCLE_COUNTER
Cycle counter.

The global clock signal or signals, clock the cycle counter. Therefore, the cycle counter is not incremented during periods of HCG when the clocks are stopped.

7.3 HN-F performance events

The HN-F performance analysis counters are used to monitor cache behavior.

For a particular cache, the cache miss rate 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.

7.3.1 Cache performance

Cache performance events are required to calculate the cache miss rate and the cache allocation.

HN-Fs support MPAM-related PMU events. See the *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A* for more information about configuration.

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. A miss results from a first-time lookup and is high priority.

PMU_HNSLC_SF_CACHE_ACCESS_EVENT

The total number of cache accesses. An access is first-time and high priority.

The performance counter architecture allows up to four HNs to collect the cache miss rate for each DTC domain. In a system with multiple DTC domains, more than four HNs can collect the cache miss rate. However, because of the CMN-700 microarchitecture, the cache miss rate that is measured at one HN-F within an SCG is a good proxy for the cache miss rate of the remaining HN-Fs.

Calculate the cache miss rate as follows:

Figure 7-1: Cache miss rate

$$\text{Cache miss rate (\%)} = \frac{\text{Total cache misses}}{\text{Total cache accesses}} \times 100$$

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 SLC hit, PMU_HNSLC_SF_CACHE_ACCESS_EVENT is only counted the first-time cache lookup is performed. The tag update is not counted as a cache access. Similarly, for Write-Back or Write*Unique transactions with an SLC allocate hint, only the first instance of an SLC lookup is counted as an access and hit or miss. The eventual victim selection and cache fill are not counted as further accesses.

Cache allocations

The cache allocation event counts the number of times an HN-F SLC 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 SLC cache.

All cache line writes, that is, Write*Unique, Write-Back, and evictions that are allocated in SLC cache, are counted towards this event.

7.3.2 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 wait period indicates whether a lack of credits is causing the bottlenecks, and also shows if the latency of requests is very high.

Calculate the message retry rate as follows:

Figure 7-2: HN-F message retry rate

$$\text{HN-F message retry rate (\%)} = \frac{\text{HN-F total messages retried}}{\text{HN-F total messages received}} \times 100$$

7.3.3 SF events

There are three snoop events that can be counted.

The following sections describe the SF performance events.

SF miss rate

This event can be used to measure the Snoop Filter efficiency.

PMU_HN_SF_HIT_EVENT

Measures the number of SF hits.

Calculate the SF hit rate as follows:

Figure 7-3: SF hit rate

$$\text{Snoop filter hit rate (\%)} = \frac{\text{Total snoop filter hits}}{\text{Total SLC lookups}} \times 100$$

SF accesses are only counted for first-time lookups, and not for the victim selection accesses or SF fills. Because the SLC lookup and SF lookups are parallel, the SLC lookups can be used to calculate the SF hit rate.

SF evictions

This event measures the frequency of SF evictions.

PMU_HN_SF_EVICTIONS_EVENT

Measures the number of SF evictions when cache invalidations are initiated.

Snoops sent and received with hit rate

These events measure the amount of shared data across clusters for a specific application, using snoop 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:

Figure 7-4: Sent and received snoops rate

$$\text{Shared data (\%)} = \frac{\text{Total snoops broadcast}}{\text{Total snoops sent}} \times 100$$

The number of broadcast and total snoops measures the shared data invalidations.

7.3.4 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:

Figure 7-5: MC message retry rate

$$\text{MC message retry rate (\%)} = \frac{\text{MC total messages retried}}{\text{MC total messages received}} \times 100$$

7.3.5 Snoop events related to SF clustering

Certain HN-F PMU events can be used to understand the performance impact of SF clustering.

The following events can be counted:

PMU_HN_SNP_SENT_CLUSTER_EVENT

Counts the number of snoops that are sent at the level of a whole cluster. This event does not count individual snoops within a cluster. For example, in a cluster with four RN-Fs, if HN-F sends four snoops, since all four snoops are sent to the same cluster they are counted as one.

PMU_HN_SF_IMPRECISE_EVICT_EVENT

Counts the number of times an evict operation from an RN does not clear the SF tracking because the line was in shared state (imprecise). If there is a single RN-F in the cluster, then the evict operation always clears the tracking for that RN-F. However, in SF clusters, the evict operation must not clear the SF tracking as other RN-Fs might still be accessing this line.

PMU_HN_SF_EVICT_SHARED_LINE_EVENT

Counts the number of times an SF eviction happened to a cache line that was in shared state. This event can be helpful in understanding of the impact of SF pollution in clustered mode.

7.3.6 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.

7.3.7 HN-F PMU event summary

The following table shows a summary of the HN-F PMU events.

Table 7-8: HN-F events

Number	Name	Description
1	PMU_HN_CACHE_MISS_EVENT	Counts total cache misses in first lookup result (high priority)
2	PMU_HNSLC_SF_CACHE_ACCESS_EVENT	Counts number of cache accesses in first access (high priority)
3	PMU_HN_CACHE_FILL_EVENT	Counts total allocations in HN SLC (all cache line allocations to SLC)
4	PMU_HN_POCQ_RETRY_EVENT	Counts number of retried requests
5	PMU_HN_POCQ_REQS_RECVD_EVENT	Counts number of requests that HN receives
6	PMU_HN_SF_HIT_EVENT	Counts number of SF hits
7	PMU_HN_SF_EVICTIONS_EVENT	Counts number of SF eviction cache invalidations initiated
8	PMU_HN_DIR_SNOOPS_SENT_EVENT	Counts number of directed snoops sent (not including SF back invalidation)
9	PMU_HN_BRD_SNOOPS_SENTEVENT	Counts number of multicast snoops sent (not including SF back invalidation)
10	PMU_HN_SLC_EVICTION_EVENT	Counts number of SLC evictions (dirty only)
11	PMU_HN_SLC_FILL_INVALID_WAY_EVENT	Counts number of SLC fills to an invalid way
12	PMU_HN_MC_RETRIES_EVENT	Counts number of retried transactions by the MC
13	PMU_HN_MC_REQS_EVENT	Counts number of requests that are sent to MC
14	PMU_HN_QOS_HH_RETRY_EVENT	Counts number of times a HighHigh priority request is protocol-retried at the HN-F
15	PMU_HNF_POCQ_OCCUPANCY_EVENT	Counts the POCQ occupancy in HN-F. Occupancy filtering is programmed in pmu_occup1_id
16	PMU_HN_POCQ_ADDRHAZ_EVENT	Counts number of POCQ address hazards on allocation
17	PMU_HN_POCQ_ATOMICS_ADDRHAZ_EVENT	Counts number of POCQ address hazards on allocation for atomic operations
18	PMU_HN_LD_ST_SWP_ADQ_FULL_EVENT	Counts number of times ADQ is full for Ld/St/SWP type atomic operations while POCQ has pending operations
19	PMU_HN_CMP_ADQ_FULL_EVENT	Counts number of times ADQ is full for CMP type atomic operations while POCQ has pending operations
20	PMU_HN_TXDAT_STALL_EVENT	Counts number of times HN-F has a pending TXDAT flit but no credits to upload
21	PMU_HN_TXRSP_STALL_EVENT	Counts number of times HN-F has a pending TXRSP flit but no credits to upload
22	PMU_HN_SEQ_FULL_EVENT	Counts number of times requests are replayed in SLC pipe because of SEQ being full
23	PMU_HN_SEQ_HIT_EVENT	Counts number of times a request in SLC hit a pending SF eviction in SEQ
24	PMU_HN_SNP_SENT_EVENT	Counts number of snoops sent including directed, multicast, and SF back invalidation
25	PMU_HN_SFBI_DIR_SNP_SENT_EVENT	Counts number of times directed snoops were sent because of SF back invalidation

Number	Name	Description
26	PMU_HN_SFBI_BRD_SNP_SENT_EVENT	Counts number of times multicast snoops were sent because of SF back invalidation.
27	PMU_HN_SNP_SENT_UNTRK_EVENT	Counts number of times snoops were sent because of untracked RN-Fs
28	PMU_HN_INTV_DIRTY_EVENT	Counts number of times SF back invalidation resulted in dirty line intervention from the RN
29	PMU_HN_STASH_SNP_SENT_EVENT	Counts number of times stash snoops were sent
30	PMU_HN_STASH_DATA_PULL_EVENT	Counts number of times stash snoops resulted in data pull from the RN
31	PMU_HN_SNP_FWDED_EVENT	Counts number of times data forward snoops were sent
32	PMU-HN_ATOMIC_FWD_EVENT	Counts number of time atomic data was forwarded between POC entries
33	PMU_HN_MPAM_REQ_OVER_HARDLIM_EVENT	Counts number of times write request cannot allocate SLC because of being over hard limit
34	PMU_HN_MPAM_REQ_OVER_SOFTLIM_EVENT	Counts number of times write request is above soft limit
35	PMU_HN_SNP_SENT_CLUSTER_EVENT	Counts number of snoops sent to clusters excluding individual snoops within a cluster
36	PMU_HN_SF_IMPRECISE_EVICT_EVENT	Counts number of times an evict operation was dropped because of SF clustering
37	PMU_HN_SF_EVICT_SHARED_LINE_EVENT	Counts number of times a shared line was evicted from SF
38	PMU_HN_POCQ_CLASS_OCCUPANCY_EVENT	Counts the given POCQ occupancy for a given class in HN-F. Class occupancy filtering is programmed in pmu_class_occup_id.
39	PMU_HN_POCQ_CLASS_RETRY_EVENT	Counts number of retried requests for a given class. Class filtering is programmed in pmu_class_occup_id.
40	PMU_HN_CLASS_MC_REQS_EVENT	Counts number of requests sent to MC for a given class. Class filtering is programmed in pmu_class_occup_id.
41	PMU_HN_CLASS_PCRDGNT_BELOW_CONDMIN_EVENT	Counts number of protocol credit grants for a given class when it is above dedicated and below conditional min. Class filtering is programmed in pmu_class_occup_id.
42	PMU_HN_NUM_SN_CBUSY_THROTTLE_EVENT	Counts number of times request to SN was throttled because of CBusy. Event filtering is programmed in pmu_cbusy_snthrottle_sel.
43	PMU_HN_NUM_SN_CBUSY_THROTTLE_MIN_EVENT	Counts number of times request to SN was throttled to the minimum because of CBusy. Event filtering is programmed in pmu_cbusy_snthrottle_sel.
44	PMU_HN_SF_PRECISE_TO_IMPRECISE_EVENT	Counts when number of sharers exceeds how many RNs could be precisely tracked in SF
45	PMU_HN_SNP_INTV_CLN_EVENT	Counts the number of times clean data intervened for a snoop request
46	PMU_HN_NC_EXCL_EVENT	Counts the number of times non-cacheable exclusive request arrived at HN-F
47	PMU_HN_EXCL_MON_OVFL_EVENT	counts the number of times exclusive monitor overflowed

7.4 RN-I performance events

External devices connect at an RN-I bridge.

7.4.1 Bandwidth at RN-I bridges

External devices connect at an RN-I bridge.

The following events measure bandwidth at the RN-I bridges:

- 7.4.1.1 Requested read bandwidth at RN-I bridges on page 1241.
- 7.4.1.2 Actual read bandwidth on interconnect on page 1242.
- 7.4.1.3 Write bandwidth at RN-I bridges on page 1242.

7.4.1.1 Requested read bandwidth at RN-I bridges

External devices connect to CMN-700 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.

RDataBeats_Port1
Number of RData beats, RVALID and RREADY, dispatched on port 1.

RDataBeats_Port2
Number of RData beats, RVALID and RREADY, dispatched on port 2.

Because CMOs are sent through the read channel, their responses are included in these events.

Calculate the read bandwidth as follows:

Figure 7-6: Read bandwidth calculation

$$\text{Read bandwidth} = \frac{\text{Number RDataBeats\_Port}n \times \text{AXIDataBeatSize}}{\text{Cycles}} \times \text{Frequency}$$

Where AXIDataBeatSize is the number of bytes for each AXI beat. Usually, this number is the same size as the AXI bus.

Note

If the data chunking feature is enabled, Read data bandwidth is calculated by counting the number of chunks being transferred on RData. This count is done

by looking at RCHUNKSTRB[n-1:0] signal, where every bit of RCHUNKSTRB represents 16B of data.

7.4.1.2 Actual read bandwidth on interconnect

RXDATFLITV measures the bandwidth that an RN-I bridge sends to the interconnect.

This event counts the number of received data flit requests that the bridge receives through the data channel. Therefore, this event measure the actual bandwidth that an RN-I bridge sends to the interconnect, and not the useful bandwidth the external devices can use.

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.

Calculate the actual read bandwidth as follows:

Figure 7-7: Actual read bandwidth

$$\text{Actual read bandwidth} = \frac{\text{RXDATFLITV x DataFlitSize}}{\text{Cycles}} \text{x Frequency}$$

7.4.1.3 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. Therefore, this event measures 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:

Figure 7-8: Actual write bandwidth

$$\text{Actual write bandwidth} = \frac{\text{TXDATFLITV x DataFlitSize}}{\text{Cycles}} \text{x Frequency}$$

7.4.2 Bottleneck analysis at RN-I bridges

CMN-700 provides events that observe the locations where the nodes or bridges are full, which can cause delays in the rest of the system.

This feature 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 condition can cause delays in the AXI managers.
- The number of read request replays, because of decoupling of the read request buffers and read data buffers in the RN-I system.

7.4.2.1 Request retry rate at RN-I bridges

TXREQFLITV_RETRIED monitors the efficiency of using dynamic credits in the system.

It does this task 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:

Figure 7-9: Retry rate

$$\text{Retry rate} = \frac{\text{TXREQFLITV\_RETRIED}}{\text{TXREQFLITV\_TOTAL}}$$

7.4.2.2 Read and write delays at RN-I bridges

To monitor the delays for both reads and writes, CMN-700 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 manager. This condition delays the I/O devices that connect to the AXI manager.

There are two measures that, together, can help you to isolate the source of bottlenecks in the system. These measures are: how full the trackers are, and the read and write bandwidth from the RN-I bridge to the interconnect. For example:

- Consider a situation where the read tracker of a specific RN-I bridge is full. However, the effective read bandwidth from the bridge is not close to the maximum expected. In this situation, 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. Therefore, the tracker is full for this reason.

You can also use the measure of how full the trackers are with AXI PMUs to monitor delays to the AXI managers.

The following events monitor the read and write trackers:

RRT_OCCUPANCY

All entries in the read request tracker are occupied. This event is a measure of oversubscription in the read request tracker.

WRT_OCCUPANCY

All entries in the write request tracker are occupied. This event is a measure of oversubscription in the write request tracker.

For CMN-700, when the `NUM_RD_REQ` parameter for an RN-I or RN-D node is configured to 128 or 256, the read tracker is divided into slices of 64 entries each. An ACE-Lite request is allocated into a particular read tracker slice based on the following aspects:

- A hash of the ARID value of the request
- Which of the three ACE-Lite subordinate interfaces receives the request

Therefore, in these configurations, the maximum number of outstanding same-ARID requests from the same ACE-Lite subordinate interface is 64.

The RRT_OCCUPANCY event covers the total occupancy of all read tracker slices.

7.4.3 RN-I PMU event summary

The following table shows a summary of the RN-I PMU events.

Table 7-9: 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 event measures the read bandwidth, including CMO responses.
2	PMU_RNI_RDATABEATS_P1	Number of RData beats (RVALID and RREADY) dispatched on port 1. This event measures the read bandwidth, including CMO responses.
3	PMU_RNI_RDATABEATS_P2	Number of RData beats (RVALID and RREADY) dispatched on port 2. This event measures the read bandwidth, including CMO responses.
4	PMU_RNI_RXDATFLITV	Number of RXDAT flits received. This event measures the true read data bandwidth, excluding CMOs.
5	PMU_RNI_TXDATFLITV	Number of TXDAT flits dispatched. This event measures the write bandwidth.
6	PMU_RNI_TXREQFLITV	Number of TXREQ flits dispatched. This event measures the total request bandwidth.
7	PMU_RNI_TXREQFLITV_RETRIED	Number of retried TXREQ flits dispatched. This event measures the retry rate.

Number	Name	Description
8	PMU_RNI_RRT_OCCUPANCY_SLICE0	All entries in the slice0 read request tracker are occupied. This event measures oversubscription in the read request tracker.
9	PMU_RNI_WRT_OCCUPANCY	All entries in the write request tracker are occupied. This event measures oversubscription in the write request tracker.
10	PMU_RNI_TXREQFLITV_REPLAYED	Number of replayed TXREQ flits. This event measures the replay rate.
11	PMU_RNI_WRCANCEL_SENT	Number of write data cancels sent. This event measures the write cancel rate.
12	PMU_RNI_WDATABEAT_P0	Number of WData beats (WVALID and WREADY) dispatched on port 0. This event measures write bandwidth on AXI port 0.
13	PMU_RNI_WDATABEAT_P1	Number of WData beats (WVALID and WREADY) dispatched on port 1. This event measures the write bandwidth on AXI port 1.
14	PMU_RNI_WDATABEAT_P2	Number of WData beats (WVALID and WREADY) dispatched on port 2. This event measures the write bandwidth on AXI port 2.
15	PMU_RNI_RRTALLOC	Number of allocations in the read request tracker. This event measures the read transaction count.
16	PMU_RNI_WRTALLOC	Number of allocations in the write request tracker. This event measures the write transaction count.
17	PMU_RNI_PADB_OCCUPANCY	All entries of the pre-allocated data buffer are occupied
18	PMU_RNI_RPDB_OCCUPANCY	All entries of the replay data buffer are occupied
19	PMU_RNI_RRT_OCCUPANCY_SLICE1	All entries in the slice1 read request tracker are occupied. This event measures oversubscription in the read request tracker.
20	PMU_RNI_RRT_OCCUPANCY_SLICE2	All entries in the slice2 read request tracker are occupied. This event measures oversubscription in the read request tracker.
21	PMU_RNI_RRT_OCCUPANCY_SLICE3	All entries in the slice3 read request tracker are occupied. This event measures oversubscription in the read request tracker.
22	PMU_RNI_WRT_THROTTLED	Write requests are being throttled because of retry. This event measures throttling in the write request tracker.
23	PMU_RNI_LDB_FULL	Link data buffer is full. This event measures full condition of link data buffer because of replays back pressure.
24	PMU_RNI_RRT_REG_RD_REQ_OCCUPANCY_SLICE0	All entries in the slice0 read request tracker are occupied with regular reads
25	PMU_RNI_RRT_REG_RD_REQ_OCCUPANCY_SLICE1	All entries in the slice1 read request tracker are occupied with regular reads
26	PMU_RNI_RRT_REG_RD_REQ_OCCUPANCY_SLICE2	All entries in the slice2 read request tracker are occupied with regular reads
27	PMU_RNI_RRT_REG_RD_REQ_OCCUPANCY_SLICE3	All entries in the slice3 read request tracker are occupied with regular reads
28	PMU_RNI_RRT_BURST_REQ_OCCUPANCY_SLICE0	All entries in the slice0 read request tracker are occupied with PCIe read Bursts
29	PMU_RNI_RRT_BURST_REQ_OCCUPANCY_SLICE1	All entries in the slice1 read request tracker are occupied with PCIe read Bursts
30	PMU_RNI_RRT_BURST_REQ_OCCUPANCY_SLICE2	All entries in the slice2 read request tracker are occupied with PCIe read Bursts
31	PMU_RNI_RRT_BURST_REQ_OCCUPANCY_SLICE3	All entries in the slice3 read request tracker are occupied with PCIe read Bursts
32	PMU_RNI_RRT_BURST_ALLOC	Number of PCIe read Burst allocations in the read request tracker. This event measures the PCIe read Burst transaction count.

Number	Name	Description
33	PMU_RNI_AWID_HASH	Incoming AWID hash has matched an existing outstanding write entry
34	PMU_RNI_ATOMIC_ALLOC	Number of Atomic request allocations in the write request tracker. This event measures the Atomic transaction count.
35	PMU_RNI_ATOMIC_OCCUPANCY	All entries in the write request tracker are occupied with atomic requests

7.5 SBSX performance events

The following contains SBSX performance event information.

7.5.1 Bandwidth at SBSX bridges

The following contains SBSX bridge bandwidth information.

The following events are used to measure bandwidth at the SBSX bridges:

- 7.5.1.1 Read bandwidth on interconnect at SBSX bridges on page 1246.
- 7.5.1.2 Write bandwidth at SBSX bridges on page 1247.
- 7.5.1.3 Total requested bandwidth at SBSX bridges on page 1247.

7.5.1.1 Read bandwidth on interconnect at SBSX bridges

The following contains information on read bandwidth on interconnect at SBSX bridges.

This event counts the number of received data flits at the SBSX and interconnect:

PMU_SBSX_RXDAT

Number of RXDAT flits received at XP from SBSX. This event is a measure of the read data bandwidth.

Calculate the actual read bandwidth as follows:

Figure 7-10: Actual read bandwidth

$$\text{Actual read bandwidth} = \frac{\text{PMU\_SBSX\_RXDAT} \times \text{DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

7.5.1.2 Write bandwidth at SBSX bridges

The following contains information on write bandwidth at SBSX bridges.

In a similar way to the read actual bandwidth event, this event monitors the number of data flits that the SBSX receives. Therefore, this event measures the actual write bandwidth that is received from the interconnect:

PMU_SBSX_TXDAT

Number of TXDAT flits dispatched from XP to SBSX. This event is a measure of the write bandwidth.

Calculate the write bandwidth as follows:

Figure 7-11: Actual write bandwidth

$$\text{Actual write bandwidth} = \frac{\text{PMU\_SBSX\_TXDAT} \times \text{DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

7.5.1.3 Total requested bandwidth at SBSX bridges

The following contains information on total requested bandwidth at SBSX bridges.

To improve efficiency when using PMU events and signals, this event combines the read and write bandwidth estimation in a single event. The PMU_SBSX_TXREQ_TOTAL event monitors the number of REQ flits that an SBSX bridge receives:

PMU_SBSX_TXREQ_TOTAL

Number of TXREQ flits dispatched from XP to SBSX. This event is a measure of the total request bandwidth.

Calculate the total bandwidth as follows:

Figure 7-12: Total requested bandwidth

$$\text{Total requested bandwidth} = \frac{\text{PMU\_SBSX\_TXREQ\_TOTAL x DataFlitSize}}{\text{Cycles}} \text{ x Frequency}$$


Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

7.5.2 Bottleneck analysis at SBSX bridges

The following contains information on bottleneck analysis at SBSX bridges.

CMN-700 provides events that observe the locations where the nodes or bridges are full, which can cause delays in the rest of the system. This feature enables you to monitor the current bottlenecks in the system, and checks multiple events in all CMN-700 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 cycles the bridge is forced to stall because of backpressures on AXI or CHI interface

The following events are used to measure bottlenecks at the SBSX bridges:

- 7.5 SBSX performance events on page 1246.

7.5.2.1 Request retry rate at SBSX bridges

The following contains information on the request retry rate at SBSX bridges.

RETRYACK_TXRSP monitors the efficiency of using dynamic credits in the system. It does this task by measuring the request retry rate:

RETRYACK_TXRSP

Number of RXREQ flits dispatched. This event is a measure of the retry rate. Calculate the retry rate as follows:
Retry rate = RETRYACK_TXRSP / RXREQFLITV_TOTAL

7.5.2.2 Delays at SBSX bridges because of backpressure

To analyze the delays in SBSX bridges, CMN-700 enables you to monitor the source of backpressure.

SBSX might have requests that are ready to send to the downstream AXI or ACE-Lite device, but it cannot send them because of backpressure from the downstream device. In this situation, SBSX holds the request in the *Receive Request Tracker* (RRT). This condition results in the RRT getting full and so the SBSX bridge cannot accept any new requests from RNs impacting system performance.

The following table contains events that monitor such backpressure from the downstream AXI or ACE-Lite device:

Table 7-10: AXI or ACE-Lite downstream monitoring events

Events	Description
ARVALID_NO_ARREADY	Number of cycles the SBSX bridge is stalled because of backpressure on AR channel
AWVALID_NO_AWREADY	Number of cycles the SBSX bridge is stalled because of backpressure on AW channel
WVALID_NO_WREADY	Number of cycles the SBSX bridge is stalled because of backpressure on W channel

If a mesh is congested with many DAT or RSP flits, it might not give link credits to SBSX in timely manner. This situation can cause DAT flits for reads or RSP flits for writes to be stalled in SBSX. The following table describes events monitor in such cases where SBSX bridge is not able to upload DAT/RSP flits on the mesh.

Table 7-11: CHI events monitor information

Events	Description
TXDATFLITV_NO_LINKCRD	Number of cycles the TXDAT flit in SBSX bridge is waiting for link credits
TXRSPFLITV_NO_LINKCRD	Number of cycles the TXRSP flit in SBSX bridge is waiting for link credits

7.5.2.3 Tracker occupancy analysis

To debug performance issues, more events are provided to measure occupancy of various trackers in SBSX. These trackers include the *Request Received Tracker* (RRT), *Request Dispatch Tracker* (RDT), and *Write Data Buffers* (WDB).

Read, write, and CMO transactions occupy RRT before they are dispatched on the AXI interface. When Read/CMO transactions are dispatched on AXI, they move from RRT to RDT. Writes remain on RRT until the write response is obtained from AXI interface and then deallocated from RRT. Knowing the occupancy of RRT and RDT independently can inform you better about the bottleneck source. In the PMU event register description section, RRT is called request tracker, while RDT is called AXI pending tracker.

The following table contains tracker occupancy information.

Table 7-12: Tracker occupancy information

Events	Description
RRT_RD_OCCUPANCY_CNT_OVFL	Read request tracker occupancy count overflow
RRT_WR_OCCUPANCY_CNT_OVFL	Write request tracker occupancy count overflow
RRT_CMO_OCCUPANCY_CNT_OVFL	CMO request tracker occupancy count overflow
WDB_OCCUPANCY_CNT_OVFL	WDB occupancy count overflow
RDT_RD_OCCUPANCY_CNT_OVFL	Read AXI pending tracker occupancy count overflow
RDT_CMO_OCCUPANCY_CNT_OVFL	CMO AXI pending tracker occupancy count overflow

7.5.3 SBSX PMU event summary

The following contains SBSX PMU event summary information.

See 4.3.17.18 por_sbsx_pmu_event_sel on page 1118.

7.6 HN-I performance events

The following contains HN-I performance event information.

7.6.1 Bandwidth at HN-I bridges

The following contains HN-I bridge bandwidth information.

The following events are used to measure bandwidth at the HN-I bridges:

- 7.6.1.1 Read bandwidth on interconnect at HN-I bridges on page 1250.
- 7.6.1.2 Write bandwidth at HN-I bridges on page 1251.
- 7.6.1.3 Total requested bandwidth at HN-I bridges on page 1251.

7.6.1.1 Read bandwidth on interconnect at HN-I bridges

The following contains information on read bandwidth on interconnect at HN-I bridges.

This event counts the number of received data flits at the HN-I and interconnect:

PMU_HNI_RXDAT

Number of RXDAT flits received at XP from HN-I. This event is a measure of the read data bandwidth.

Calculate the actual read bandwidth as follows:

Figure 7-13: Actual read bandwidth

$$\text{Actual read bandwidth} = \frac{\text{PMU\_HNI\_RXDAT} \times \text{DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

7.6.1.2 Write bandwidth at HN-I bridges

The following contains information on write bandwidth at HN-I bridges.

In a similar way to the read actual bandwidth event, this event monitors the number of data flits that the HN-I receives. Therefore this event measures the actual write bandwidth that is received from the interconnect:

PMU_HNI_TXDAT

Number of TXDAT flits dispatched from XP to HN-I. This event is a measure of the write bandwidth.

Calculate the write bandwidth as follows:

Figure 7-14: Actual write bandwidth

$$\text{Actual write bandwidth} = \frac{\text{PMU\_HNI\_TXDAT} \times \text{DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

7.6.1.3 Total requested bandwidth at HN-I bridges

The following contains information on total requested bandwidth at 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. The PMU_HNI_TXREQ_TOTAL event monitors the number of REQ flits that an HN-I bridge receives:

PMU_HNI_TXREQ_TOTAL

Number of TXREQ flits dispatched from XP to HN-I. This event is a measure of the total request bandwidth.

Calculate the total bandwidth as follows:

Figure 7-15: Total requested bandwidth

$$\text{Total requested bandwidth} = \frac{\text{PMU\_HNI\_TXREQ} \times \text{DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

This event is tracked in the DTM watchpoint in the XP where the component is located.

7.6.2 Bottleneck analysis at HN-I bridges

The following contains information on bottleneck analysis at HN-I bridges.

Locations where the nodes or bridges are full can cause delays in the rest of the system. CMN-700 provides events that observe locations where the nodes or bridges are full. This feature enables you to monitor the current bottlenecks in the system, and checks multiple events in all CMN-700 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 times requests are serialized because of ordering requirements
- The number of cycles the bridge is forced to stall because of backpressures

The following events are used to measure bottlenecks at the HN-I bridges:

- 7.6 HN-I performance events on page 1250.

7.6.2.1 Request retry rate at HN-I bridges

The following contains information on the request retry rate at HN-I bridges.

RETRYACK_TXRSP monitors the efficiency of using dynamic credits in the system. It does this task by measuring the request retry rate:

RETRYACK_TXRSP

Number of RXREQ flits dispatched. This event is a measure of the retry rate. Calculate the retry rate as follows:
Retry rate = RETRYACK_TXRSP / RXREQFLITV_TOTAL

7.6.2.2 Delays at HN-I bridges because of ordering requirements

When requests are received at an HN-I, there are different ordering guarantees the HN-I bridge must maintain based on the source and attributes of the request.

The requests are sometimes serialized, indicating a lower than expected bandwidth at HN-I, as the following table shows.

Table 7-13: PCIe and non-PCIe RN request information

Request	Description
NONPCIE_SERIALIZED	Number of non-PCIe RN requests that are serialized.
PCIE_SERIALIZED	Number of PCIe RN requests that are serialized.

7.6.2.3 Delays at HN-I bridges because of backpressure

To analyze the delays in HN-I bridges, CMN-700 enables you to monitor the source of backpressure.

HN-I might have requests that are ready to be sent to AXI or ACE-Lite downstream, but cannot send them due to backpressure from AXI or ACE-Lite downstream. In this situation, HN-I holds the request in the RRT. As a result, the RRT gets full. Therefore, the HN-I bridge cannot accept any new requests from RNs, impacting system performance.

This table describes the events that monitor such backpressure from AXI or ACE-Lite downstream:

Table 7-14: AXI or ACE downstream events monitor information

Events	Description
ARVALID_NO_ARREADY	Number of cycles the HN-I bridge is stalled because of backpressure on AR channel
AWVALID_NO_AWREADY	Number of cycles the HN-I bridge is stalled because of backpressure on AW channel
WVALID_NO_WREADY	Number of cycles the HN-I bridge is stalled because of backpressure on W channel

Even if the AXI or ACE-Lite downstream is ready to accept new requests, the HN-I bridge cannot send them downstream while the RDT is full. The lifetime of a request in the RDT depends on response latency from AXI or ACE-Lite downstream and backpressure on TXDAT channel.

This table describes the events that monitor cases where an HN-I bridge is unable to send new requests to AXIor ACE-Lite downstream:

Table 7-15: AXI or ACE downstream events monitor information (no new requests sent)

Events	Description
ARREADY_NO_ARVALID	Number of cycles the AR channel is waiting for new requests from HN-I bridge
AWREADY_NO_AWVALID	Number of cycles the AW channel is waiting for new requests from HN-I bridge

If the mesh is congested with many DAT flits, then there might be a delay before it gives link credits to HN-I. This delay results in the stalling of DAT flits for reads in HN-I. This table describes events that monitor cases where an HN-I bridge is not able to upload a DAT flit on the mesh.

Table 7-16: CHI events monitor information

Events	Description
TXDATFLITV_NO_LINKCRD	Number of cycles the TXDAT flit in HN-I bridge is waiting for link credits

7.6.2.4 Tracker occupancy analysis in HN-I

To debug performance issues, more events are provided to measure occupancy of various trackers in HN-I such as RRT, RDT, and WDB.

Read and write transactions occupy RRT before they are dispatched on the AXI interface. When read and write transactions are dispatched on AXI, they move from RRT to RDT. Reads and writes remain on RDT until all the responses are obtained from the AXI interface. The transactions are then deallocated from RDT. Knowing the occupancy of RRT and RDT independently can inform you better about the bottleneck source. In the PMU event register description section, RRT is called request tracker while RDT is called AXI pending tracker.

The following table contains tracker occupancy information:

Table 7-17: Tracker occupancy information

Events	Description
RRT_RD_OCCUPANCY_CNT_OVFL	Read occupancy count overflow event in RRT
RRT_WR_OCCUPANCY_CNT_OVFL	Write occupancy count overflow event in RRT
RDT_RD_OCCUPANCY_CNT_OVFL	Read occupancy count overflow event in RDT
RDT_WR_OCCUPANCY_CNT_OVFL	Write occupancy count overflow event in RDT
WDB_OCCUPANCY_CNT_OVFL	WDB occupancy count overflow event

7.6.3 HN-I PMU event summary

The following contains HN-I PMU event summary information.

See 4.3.9.22 por_hni_pmu_event_sel on page 596.

7.6.4 HN-P PMU events

HN-P has dedicated RRTs, RDTs, and WDBs for requests from RN-Is and RN-Ds connected to PCIe managers. There are associated PMU events for these resources.

For more information about the associated events, see 4.3.9.23 por_hnp_pmu_event_sel on page 598.

The por_hnp_pmu_event_sel register only outputs on the corresponding TXPMU output if por_hni_pmu_event_sel bits [5], [13], [21], and [29] are set to 0. Otherwise, the value on the HN-I PMU is available.

7.7 DN performance events

The following contains DN performance event information.

The following table shows a summary of the DN PMU events.

Table 7-18: DN PMU event summary

Number	Description
1	Number of TLBI DVM op requests
2	Number of BPI DVM op requests
3	Number of PICI DVM op requests
4	Number of VICI DVM op requests
5	Number of DVM sync requests
6	Number of DVM op requests that were filtered using VMID filtering
7	Number of DVM op requests to RN-Ds, BPI, or PICI/VICI, that are filtered.
8	Number of retried REQs
9	Number of SNPs sent to RNs
10	Number of SNPs stalled to RNs because of lack of credits
11	DVM tracker full counter
12	DVM tracker occupancy counter
6'h00	No event
6'h01	Number of TLBI DVM op requests
6'h02	Number of BPI DVM op requests
6'h03	Number of PICI DVM op requests
6'h04	Number of VICI DVM op requests
6'h05	Number of DVM sync requests
6'h06	Number of DVM op requests that were filtered using VMID filtering
6'h07	Number of DVM op requests to RN-Ds (BPI or PICI/VICI) that were filtered
6'h08	Number of retried REQ

Number	Description
6'h09	Number of SNPs sent to RNs
6'h0a	Number of SNPs stalled to RNs because of lack of Crds
6'h0b	DVM tracker full counter
6'h0c	DVM RN-F tracker occupancy counter
6'h0d	DVM CXHA tracker occupancy counter
6'h0e	DVM peer DN tracker occupancy counter
6'h0f	DVM RN-F tracker Alloc
6'h10	DVM CXHA tracker Alloc
6'h11	DVM peer DN tracker Alloc
6'h12	TXSNP stall because of number outstanding limit
6'h13	RXSNP stall starvation threshold hit
6'h14	TXSNP SYNC stall because of outstanding early completed op

The pmu_occup1_id field in the por_dn_pmu_event_sel register is used to program the occupancy counter for specific operations types. The following table summarizes the options.

Table 7-19: Field values for pmu_occup1_id

pmu_occup1_id values	Description
0b0000	All
0b0001	DVM ops
0b0010	DVM syncs


Note

In HN-D, HN-T and HN-V, DN PMU events can be accessed only when the corresponding HN-I PMU select is 0 (NONE).

DN events can be accessed through the HN-D, HN-T, and HN-V. The por_dn_pmu_event_sel register outputs on corresponding TXPMU output only if por_hni_pmu_event_sel bits [5], [13], [21], and [29] are set to 0. Otherwise the value on the HN-I PMU is available.

7.8 XP PMU event summary

The following contains XP PMU event summary information.

Each of the XP PMU events is associated with:

- One of six XP ports:
  - If using a mesh configuration, these ports can be east, west, north, south, device port P0, P1, P2, or P3, depending on the configuration.
- One of 6 CHI channels - REQ, RSP, SNP, DAT, RSP2, or DAT2. RSP2 and DAT2 are valid when dual channel mode is enabled.

Up to four XP PMU events can be specified using the por_mxp_pmu_event_sel register. For more information about this register, see 4.3.13.21 por_mxp_pmu_event_sel on page 903.

The following table shows a summary of the XP PMU events.

Table 7-20: XP PMU event summary

Number	Name	Description
1	PMU_XP_TXFLIT_VALID	Number of flits that are transmitted on a specified port and CHI channel. This event measures the flit transfer bandwidth from an XP. **Note:** On device ports, this event also includes link flit transfers.
2	PMU_XP_TXFLIT_STALL	Number of cycles when a flit is stalled at an XP waiting for link credits at a specified port and CHI channel. This event measures the flit traffic congestion on the mesh and at the flit download ports.

7.9 CCG performance events

The following contains CCG performance event information.

CCG consists of RA, HA, LA, and an optional RN-I. PMU events from all these components are muxed out to the shared 4-bit wide PMU event bus connected to the respective XP. Therefore a total of four events can be counted simultaneously from any given CCG.

When events from multiple of these subunits (RA, HA, LA, and RN-I) are selected simultaneously, the priority of event selection is HA > RA > LA > RN-I. For a lower priority event to be selected, the *pmu_event_sel bit of the higher priority subunit must be set to zero. For example, event selection of RA, HA, and LA must be set to zero for RN-I events to be selected.

Table 7-21: RA event

Subunit	Event select	Name	Description
RA	0x00	No event	RA NULL event
	0x41	RA_REQ_TRK_OCC	Request Tracker (RHT) occupancy count overflow
	0x42	RA_SNP_TRK_OCC	Snoop Tracker (RHT) occupancy count overflow
	0x43	RA_RD_DAT_BUF_OCC	Read Data Buffer (RDB) occupancy count overflow
	0x44	RA_WR_DAT_BUF_OCC	Write Data Buffer (WDB) occupancy count overflow
	0x45	RA_SNP_SINK_BUFF_OCC	Snoop Credit Buffer (SSB) occupancy count overflow
	0x46	RN_SNP_BCASTS	Number of inbound broadcast snoops received
	0x47	RA_REQ_CHAINS	Number of request chains formed larger than one
	0x48	RA_REQ_CHAIN_AVG_LEN	Average size of request chains, only for chains size larger than one
	0x49	RA_CHI_INT_RSP_UPLOADS_STALLS	RA internal CHI response upload stalls because of contention with HA
	0x4A	RA_CHI_INT_DAT_UPLOAD_STALLS	RA internal CHI data upload stalls because of contention with HA

Subunit	Event select	Name	Description
	0x4B	RA_DAT_PCRD_STALLS_LNK0	Memory Data Request available, but no DAT credits to send over protocol LinkEnd0
	0x4C	RA_DAT_PCRD_STALLS_LNK1	Memory Data Request available, but no DAT credits to send over protocol LinkEnd1
	0x4D	RA_DAT_PCRD_STALLS_LNK2	Memory Data Request available, but no DAT credits to send over protocol LinkEnd2
	0x4E	RA_REQ_PCRD_STALLS_LNK0	Memory Request available, but no REQ credits to send over protocol LinkEnd0
	0x4F	RA_REQ_PCRD_STALLS_LNK1	Memory Request available, but no REQ credits to send over protocol LinkEnd1
	0x50	RA_REQ_PCRD_STALLS_LNK2	Memory Request available, but no REQ credits to send over protocol LinkEnd2
	0x51	RA_CHI_EXT_RSP_UPLOADS_STALLS	CHI response upload stalls because of mesh backpressure
	0x52	RA_CHI_EXT_DAT_UPLOAD_STALLS	CHI data upload stalls because of mesh backpressure
	0x53	RA_MISC_PCRD_STALLS_LNK0	Not applicable and must not be set
	0x54	RA_MISC_PCRD_STALLS_LNK1	Not applicable and must not be set
	0x55	RA_MISC_PCRD_STALLS_LNK2	Not applicable and must not be set
	0x56	RA_REQ_TRK_ALLOC	Request Tracker (RHT) allocation
	0x57	RA_SNP_TRK_ALLOC	Snoop Tracker (RHT) allocation
	0x58	RA_RD_DAT_BUF_ALLOC	Read Data Buffer (RDB) allocation
	0x59	RA_WR_DAT_BUF_ALLOC	Write Data Buffer (WDB) allocation
	0x5A	RA_SNP_SINK_BUFF_ALLOC	Snoop Credit Buffer (SSB) allocation

Table 7-22: HA events

Subunit	Event select	Name	Description
HA	0x00	No event	HA NULL event
	0x61	HA_RD_DAT_BYPASS	Read data bypass taken
	0x62	HA_CHI_RSP_UPLOAD_STALLS	HA internal CHI response upload stalls because of contention with RA
	0x63	HA_CHI_DAT_UPLOAD_STALLS	HA internal CHI data upload stalls because of contention with RA
	0x64	HA_SNP_PCRD_STALLS_LNK0	Snoop Request available, but no SNP credits to send over protocol LinkEnd0
	0x65	HA_SNP_PCRD_STALLS_LNK1	Snoop Request available, but no SNP credits to send over protocol LinkEnd0
	0x66	HA_SNP_PCRD_STALLS_LNK2	Snoop Request available, but no SNP credits to send over protocol LinkEnd0
	0x67	HA_REQ_TRK_OCC	Request Tracker occupancy count overflow

Subunit	Event select	Name	Description
	0x68	HA_RD_DAT_BUF_OCC	*Read Data Buffer* (RDB) occupancy count overflow
	0x69	HA_RD_DAT_BUF_BYP_OCC	Read data response bypassing RDB occupancy count overflow
	0x6A	HA_WR_DAT_BUF_OCC	Write data buffer (WDB) occupancy count overflow
	0x6B	HA_SNP_TRK_OCC	Snoop Tracker occupancy count overflow
	0x6C	HA_SNP_DAT_BUF_OCC	Snoop data buffer (SDB) occupancy count overflow
	0x6D	HA_SNP_HAZ_OCC	Snoop Hazard Buffer occupancy count overflow
	0x6E	HA_REQ_TRK_ALLOC	Request Tracker allocation
	0x6F	HA_RD_DAT_BUF_ALLOC	Read data buffer (RDB) allocation
	0x70	HA_RD_DAT_BUF_BYP_ALLOC	Number of Read data responses bypassing RDB
	0x71	HA_WR_DAT_BUF_ALLOC	Write data buffer (WDB) allocation
	0x72	HA_SNP_TRK_ALLOC	Snoop Tracker allocation
	0x73	HA_SNP_DAT_BUF_ALLOC	Snoop data buffer (SDB) allocation
	0x74	HA_SNP_HAZ_ALLOC	Snoop Hazard Buffer allocation
	0x75	PB_RHU_REQ_OCC	Passive Request Buffer RHU request occupancy count overflow
	0x76	PB_RHU_REQ_ALLOC	Passive Request Buffer RHU request allocation count overflow
	0x77	PB_RHU_PCIE_REQ_OCC	Passive Request Buffer RHU PCIe Streaming/Reserved request occupancy count overflow
	0x78	PB_RHU_PCIE_REQ_ALLOC	Passive Request Buffer RHU PCIe Streaming/Reserved request allocation count overflow
	0x79	PB_PCIE_WR_REQ_OCC	Passive Request Buffer CCG_RNI PCIe write request occupancy count overflow
	0x7A	PB_PCIE_WR_REQ_ALLOC	Passive Request Buffer CCG_RNI PCIe write request allocation count overflow
	0x7B	PB_PCIE_REG_REQ_OCC	Passive Request Buffer CCG_RNI PCIe read request occupancy count overflow
	0x7C	PB_PCIE_REG_REQ_ALLOC	Passive Request Buffer CCG_RNI PCIe read request allocation count overflow

Subunit	Event select	Name	Description
	0x7D	PB_PCIE_RSVD_REQ_OCC	Passive Request Buffer CCG_RNI PCIe GIC/TCU read request occupancy count overflow
	0x7E	PB_PCIE_RSVD_REQ_ALLOC	Passive Request Buffer CCG_RNI PCIe GIC/TCU read request allocation count overflow
	0x7F	PB_RHU_DAT_OCC	Passive Data Buffer RHU data occupancy count overflow
	0x80	PB_RHU_DAT_ALLOC	Passive Data Buffer RHU data allocation count overflow
	0x81	PB_RHU_PCIE_DAT_OCC	Passive Data Buffer RHU PCIe Streaming/Reserved write data occupancy count overflow
	0x82	PB_RHU_PCIE_DAT_ALLOC	Passive Data Buffer RHU PCIe Streaming/Reserved write data allocation count overflow
	0x83	PB_PCIE_WR_DAT_OCC	Passive Data Buffer CCG_RNI write data occupancy count overflow
	0x84	PB_PCIE_WR_DAT_ALLOC	Passive Data Buffer CCG_RNI write data allocation count overflow

Note

CCG-HA's PB allocation events use a counter width of 4. This causes the following events to be triggered once for every 16 requests allocated:

- 8'h76: CCHA_PMU_EVENT_PB_RHU_REQ_ALLOC
- 8'h78: CCHA_PMU_EVENT_PB_RHU_PCIE_REQ_ALLOC
- 8'h7A: CCHA_PMU_EVENT_PB_PCIE_WR_REQ_ALLOC
- 8'h7C: CCHA_PMU_EVENT_PB_PCIE_REG_REQ_ALLOC
- 8'h7E: CCHA_PMU_EVENT_PB_PCIE_RSVD_REQ_ALLOC

Table 7-23: LA events

Subunit	Event select	Name	Description
LA	0x00	No event	LA NULL event
	0x21	LA_RX_CXS	Number of RX CXS flits
	0x22	LA_TX_CXS	Number of TX CXS flits
	0x23	LA_RX_CXS_AVG_SIZE	Average size of RX CXS flits. Gives the packing efficiency of inbound CXS flits
	0x24	LA_TX_CXS_AVG_SIZE	Average size of TX CXS flits. Gives the packing efficiency of outbound CXS flits
	0x25	LA_TX_CXS_LCRD_BACKPRESSURE	CXS backpressure because of lack of CXS credits

Subunit	Event select	Name	Description
	0x26	LA_LINK_CRDBUF_OCC	LA RX RAM buffer occupancy count overflow
	0x27	LA_LINK_CRDBUF_ALLOC	LA RX RAM buffer allocation

7.10 Occupancy and lifetime measurement using PMU events

CMN-700 has PMU events to measure the average occupancy of a tracker and measure the average lifetime of the requests in that tracker.

This event is implemented for many of the trackers in CMN-700 units (HN-F, RN-I, RN-D, HN-I, and others). The following formula measures the average occupancy and lifetime and can be applied to all the trackers where this event is supported:

Occupancy measurement

The formula to measure the occupancy is:

Figure 7-16: Average occupancy

$$\text{Average Occupancy (entries)} = \frac{\text{PMU\_OCCUPANCY\_EVENT} << 12}{\text{PMU\_CYCLE\_COUNTER}}$$

For example, for RN-I RRT average occupancy, the formula is:

Figure 7-17: Average RRT occupancy

$$\text{Average RRT Occupancy (entries)} = \frac{\text{PMU\_RNI\_RRT\_OCCUPANCY\_EVENT} << 12}{\text{PMU\_CYCLE\_COUNTER}}$$

Lifetime measurement

If a tracker supports lifetime event, the formula to measure the lifetime is:

Figure 7-18: Average lifetime

$$\text{Average Lifetime (cycles)} = \frac{\text{PMU\_OCCUPANCY\_EVENT} << 12}{\text{PMU\_NUM\_TRACKER\_ALLOCATIONS}}$$

For example, for RN-I RRT average lifetime, the formula is:

Figure 7-19: Average RRT lifetime

$$\text{Average Lifetime (cycles)} = \frac{\text{PMU\_RNI\_RRT\_OCCUPANCY} << 12}{\text{PMU\_RNI\_RRTALLOC}}$$


Note

Above calculations assume a counter width of 12, which is valid for all units other than CCG. CCG units use a counter width of 8 and so the measurements need to be updated accordingly for CCG-RA, CCG-HA, CCLA.

HN-F supports collecting occupancy according to the request types. The following table describes the opcode filtering types that are supported.

Table 7-24: Supported opcode filtering types

pmu_occup1_id	Opcode type
0b000	All request types
0b001	Read request types
0b010	Write request types
0b011	Atomic request types
0b100	Stash request types

When filtering is enabled, pmu_occup1_id must return to the default value to collect occupancy for all request types.

7.11 DEVEVENT

CMN-700 HN-Fs support device-specific events that are together called DEVEVENT. These events are sent along with the completion of a transaction.

Completion of a transaction can be a data response (DAT) or completion response (RSP). These events contain information regarding the transaction encountering SLC hit or miss. And it also includes information about snoops sent to resolve coherency actions. These events can be measured using watchpoints on the XP that the RN-F is connected. For information about watchpoint usage, see 6.1.1 DTM watchpoint on page 1202.

The following table shows the DEVEVENT encodings from HN-F.

Table 7-25: DEVEVENT encodings from HN-F

Encoding	Description
2'b00	Line missed in SLC and no snoops sent
2'b01	Line missed in SLC and directed snoop sent

Encoding	Description
2'b10	Line missed in SLC and broadcast snoops sent
2'b11	Line hit in SLC and no snoops sent

DEVEVENT field is sent across CML_SMP link for data and completion responses originating from remote HN-F. Responses from other CMN-700 devices have the default 2'b00 as the DEVEVENT value.

Appendix A Protocol feature compliance

This appendix describes the features that CMN-700 implements from different protocol and architecture specifications.

A.1 AXI and ACE-Lite feature support

AXI and ACE-Lite provides various optional features through interface properties. CMN-700 supports some of these properties and whether a property is supported depends on the node type.

The following table shows the AXI and ACE-Lite properties that the different CMN-700 nodes with AXI and ACE-Lite interfaces support.

Table A-1: AXI and ACE-Lite feature support

AXI or ACE-Lite property	Support		
	RN-I	HN-I	SBSX
DVM_v8 and DVM_v8.1	Y (RN-D)	N	N
Multi_Copy_Atomicity	Y	Y	Y
Ordered_Write_Observation	Y	N	N
WriteEvict_Transaction	N	N	N
Atomic_Transactions	Y	N	N
Barrier_Transactions	N	N	N
Cache_Stash_Transactions	Y	N	N
Check_Type[1]	Y	Y	Y
Coherency_Connection_Signals	Y	N	N
DeAllocation_Transactions	Y[2]	N	N
Loopback_Signals	Y	N	N
NSAccess_Identifiers	Y	N	Y
Persist_CMO	Y	N	Y
Poison	Y	Y	Y
QoS_Accept	N	N	N
Trace_Signals	Y	Y	Y
Untranslated_Transactions	N	N	N
Wakeup_Signals	Y	Y	Y
CMO_On_Read	Y	N	Y
CMO_On_Write	Y	N	Y
MPAM_Support	Y	Y	Y

[1] Value becomes Odd_Parity_Byte_Data when global `DATACHECK_EN` parameter is TRUE
[2] ReadOnceMakeInvalid or ReadOnceCleanInvalid

AXI or ACE-Lite property	Support		
	RN-I	HN-I	SBSX
Read_Data_Chunking	Y	N	N
Read_Interleaving_Disabled[3]	Y	N	N
Unique_ID_Support	Y	Y	Y
Consistent_DECERR	Y	Y	Y
DVM_Message_Support	Y	N	N
DVM_v8.4	Y (RN-D)	N	N
DVM_On_Read	Y (RN-D)	N	N
DVM_On_Snoop	Y (RN-D)	N	N
Exclusive_Accesses	Y	N	N
Fixed_Burst	Y	N	N
Max_Transaction_Bytes	4KB	64B for HN-I/D/T/V & any for HN-P	64B
MTE_Support	Basic	False	Standard
Prefetch_Transaction	N	N	Y
Regular_Transactions_Only	N	N	N
Shareable_Transactions	Y	N	N
Write_Plus_CMO	Y	N	Y
WriteZero_Transaction	N	N	Y

A.2 CHI feature support

CHI provides various optional features through interface properties. CMN-700 supports some of these properties.

The following table shows the CHI properties that CMN-700 supports.

Table A-2: CHI feature support

CHI property	Support	Comments
Atomic_Transactions	Y	-
Cache_Stash_Transactions	Y	-
Direct_Memory_Transfer	Y	-
Direct_Cache_Transfer	Y	-
Data_Poison	Y	-
Data_Check	Y	-
Check_Type[4]	Y	Odd parity checking for data signals only.
CleanSharedPersistSep_Request	Y	-

[3] HN-I and SBSX can support both interleaving and no interleaving
[4] Value becomes Odd_Parity_Byte_Data when global `DATACHECK_EN` parameter is TRUE

CHI property	Support	Comments
MPAM_Support	Y	-
CCF_Wrap_Order	N	True for most of the nodes, but not all
Req_Addr_Width	Y	-
NodeID_Width	Y	Supported values are 7-11
Data_Width	N	Fixed to 256
Enhanced_Features	Y	Support enabled for all enhanced features
MTE_Support	Y	Memory tagging is supported
DVM_v8	Y	
DVM_v8.1	Y	
DVM_v8.4	Y	

A.3 CXS property support

The following provides information on CXS properties and their relevant support.

Table A-3: CXS-B interface property support

CXS property (TX and RX)	Support	Comment
CXSDATAFLITWIDTH	512	-
CXSMAXPKTPERFLIT	2	Only one packet per CXS flit
CXSCONTINUOUSDATA	TRUE	Do not care, can be connected to TRUE or FALSE
CXSERRORFULLPKT	TRUE	Do not care, can be connected to TRUE or FALSE
CXSCHECKTYPE	None	-
CXSLINKCONTROL	Explicit_Credit_Return	-
CXS_PROTOCOL_TYPE	TRUE	-
CXS_LAST	TRUE	-

A.4 CHI feature support for CML

The following provides CHI information for CML support.

The following table contains CHI support for CML settings.

Table A-4: CHI support for CML

CHI feature	CML support		Comments
	Local	Remote	
Coherency	Yes	Yes	-
Ordering	Yes	Yes	-

CHI feature	CML support		Comments
	Local	Remote	
Atomics	Yes	Yes	-
Exclusive Accesses	Yes	Yes	For more details on CML handling of Exclusive accesses, see 3.9.2 Exclusive accesses on page 228
Cache Stashing	Yes	SMP mode only*	Remote support: Stash* request are sent to the remote chip over CML_SMP link. Stash*ID fields are not passed through along with the request. SnpStash* are not supported over CML_SMP link.
DVM Operations	Yes	SMP mode only	-
Error Handling			
- Response Error	Yes	Yes	-
- Data Check	Yes	No	-
- Poison	Yes	Yes	Mandatory for CMN-700
QoS			
- Request	Yes	Yes	-
- Snoop	Yes	No	-
Data return from Shared Clean	Yes	No	-
Direct Cache Transfer (DCT)	Yes	No	Local support includes local RN-F sending data directly to CML gateway block
Direct Memory Transfer (DMT)	Yes	No	Local support includes local SN-F sending data directly to CML gateway block
I/O Deallocation transactions	Yes	Yes	-
CleanSharedPersist CMO	Yes	Yes	-
CleanSharedPersistSep CMO	Yes	SMP mode only	-
Prefetch Target	Yes	Yes	-
Trace Tag	Yes	SMP mode only	-
System Coherency Interface (SYSCOREQ and SYCOACK)	Yes	Yes (using s/w bits)	-
Partial cache state	Yes	No	CCRA, inside CML block, does not accept the following requests and responses: WriteBackPtl, WriteCleanPtl, SnpRespDataPtl*
Streaming and Optimized Streaming of Ordered Writes	Yes	SMP mode only	Remote support: In SMP mode, high BW streaming of ordered writes from PCIe RN-I or RN-D can be enabled through CCG
MPAM	Yes	SMP mode only	-
MakeReadUnique (MRU)	Yes	SMP mode only	Remote support: Both Exclusive and Non-exclusive

CHI feature	CML support		Comments
	Local	Remote	
StashOnceSep	Yes	SMP mode only	-
Write*Zero	Yes	SMP mode only	-
Read PerferUnique	Yes	SMP mode only	-
Combined Write+CMO	Yes	SMP mode only*	* Sent as separate write and CMO ops over CML_SMP link
SLC Replacement	Yes	SMP mode only	-
Deep attribute	Yes	SMP mode only	-
SnpQuery	Yes	SMP mode only	-
SnpPerferUnique	Yes	SMP mode only	-
Memory Tagging Extensions	Yes	No	MTE not supported over SMP link
CBusy	Yes	SMP mode only	See 3.9.5 Completer Busy indication on page 235
CleanSharedPersistSep	Yes	SMP mode only	-


Note

For a given cacheline address, CHI protocol requires an interconnect (ICN) to not send Snoops until CompAck or Data are received at HN-F. However, there are exceptions to this requirement in systems with CML enabled in the following circumstance.

If the CML gateway block receives a WriteClean. This exception happens because the CML gateway block synthesizes the Snoop here on a CHIWriteClean, and it assumes that the RN-F must already see completion for prior request before RN-F dispatches WriteClean. This exception is acceptable because RN-F has already issued WriteClean for the same cacheline. It cannot do that unless the prior request to the same cacheline has received completion.

A.5 CXL support

The following provides CXL information for CML support.

CMN-700 supports CXL.mem(Type3) functionality for host implementation and is compliant with *Compute Express Link (CXL) specification revision 2.0 dated October 2020* with the following limitations:

- The following optional features are not supported
  - CXL.cachemem IDE feature
  - CXL.cachemem Error Isolation
- QoS telemetry: Incoming CXL DevLoad is mapped to CHI Cbusy. See 3.9.5 Completer Busy indication on page 235
- CXL registers implemented in CCG can be accessed through the corresponding CCG's APB port. List of registers implemented in CCG and their corresponding offsets can be found in section 4.3 CXL APB Register description. Same registers are implemented in CCLA register space and these can be accessed using CMN configuration read/write accesses. By default access to these CXL registers is provided through CMN configuration mechanism and S/W must set enable_cxl_regs_apb_access bit in por_ccla_aux_ctl register to allow access through the APB port. All the other registers as defined by CXL specification are expected to be implemented in CXL controller IPor SoC.
- System software is expected to map the contents from HDM decoder to CMN-700 HN SAM and RN SAM structures. This programming is required to route CXL.mem traffic to appropriate CXL2.0 memory device. Host HDM decoder structures, one per CHBCR space as defined by CXL specification, are expected to be implemented outside CMN-700.
- In bound viral from CXL device is terminated at the corresponding gateway block and can be optionally reported as CMN-700 uncorrectable error (UE).
- MemSpecRead is supported
- DevLoad is supported
- APB port is provided for external controller IP to access CXL registers implemented in CCG
- CMN-700 supports a maximum of 16 CCGs for CXL type3 device attachment

Appendix B Signal descriptions

This appendix describes the external I/O signals that CMN-700 implements for connection to other hardware in the system.

B.1 About the signal descriptions

CMN-700 signals are composed of a base name along with identifiers that indicate unique product configuration.

Because there are multiple identical interfaces in CMN-700, the signal names that this appendix describes are often only root names. The actual signal name includes a port-specific identifier suffix.

The system configuration determines which of the signals are used in a particular system.


Note

Unless specified otherwise, CMN-700 signals are active-HIGH.

B.2 ACE-Lite and AXI Interface signals

CMN-700 interfaces use RootName as the signal name within a more fully specified convention.

All signal names in this section consist of a root name, RootName. CMN-700 interfaces use RootName within a more fully specified signal name as follows:

- CMN-700 ACE-Lite and AXI interface signal name == RootName_[S|M]<#a>_NID#b, where:


Note

S | M	Defines either a subordinate or manager 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.

B.2.1 ACE-Lite-with-DVM subordinate interface signals

This interface is present as the ACE-Lite-with-DVM subordinate port for an RN-D bridge.

Signal definitions

Table B-1: ACE-Lite-with-DVM subordinate interface signals

Signal	Direction	Description	Connection information
ACLKEN_S	Input	AXI bus clock enable	Connect to clock enable logic. Tie HIGH if RN-I port is unused.
ACWAKEUP_S	Output	Indication that the interconnect is starting a transaction that is being sent to the DVM manager (SMMU)	Connect to corresponding manager device, if populated
AWAKEUP_S	Input	Indication that the manager is starting a transaction that is being sent to the interconnect	Connect to corresponding manager device, if populated, otherwise tie LOW
RNID_SAM_STALL_DIS	Input	Disables RN SAM programming stall for specified RN	Tie HIGH if boot programming, including RN SAM, is done through this RN-I port. Otherwise, tie LOW.
SYSCOREQ_S	Input	Request to enter DVM domain when asserted and to exit the DVM domain when deasserted. SYSCOREQ and SYSCOACK implement a four-phase handshake protocol.	Connect to corresponding manager device. Tie LOW if manager is not populated or does have port.
SYSCOACK_S	Output	Acknowledge for DVM domain entry or exit	Connect to corresponding manager device, if populated
AWREADY_S	Output	Write address ready	Connect to corresponding manager device, if populated
AWVALID_S	Input	Write address valid	Connect to corresponding manager device, if populated, otherwise tie LOW
AWID_S[10:0]	Input	Write address ID	
AWADDR_S[n:0]	Input	Write address The value of `n` is configuration-dependent.	
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[n:0]	Input.	User-defined signal	
AWPROT_S[2:0]	Input	Write protection type `n` = (`REQ_RSVDC_WIDTH` + `RSVDC_PBHA_WIDTH` (if enabled) + `RSVDC_STRONGNC_EN` (if enabled) -1)	
AWQOS_S[3:0]	Input	Write *Quality of Service* (QoS) identifier	
AWSNOOP_S[3:0]	Input	Write transaction type	
AWDOMAIN_S[1:0]	Input	Write Shareability domain	
AWATOP_S[5:0]	Input	Atomic operation	Connect to corresponding manager device, if populated, otherwise tie LOW
AWSTASHNID_S[10:0]	Input	Indicates the node identifier of the physical interface that is the target interface for the cache stash operation	Connect to corresponding manager device, if populated, otherwise tie LOW

Signal	Direction	Description	Connection information
AWSTASHNIDEN_S	Input	When asserted, indicates that the AWSTASHNID signal is valid and should be used	Connect to corresponding manager device, if populated, otherwise tie LOW
AWSTASHLPID_S[4:0]	Input	Indicates the logical processor subunit that is associated with the physical interface that is the target for the cache stash operation	Connect to corresponding manager device, if populated, otherwise tie LOW
AWSTASHLPIDEN_S	Input	When asserted, indicates that the AWSTASHLPID signal is enabled and should be used	Connect to corresponding manager device, if populated, otherwise tie LOW
AWTRACE_S	Input	Trace signal that is associated with the AW Write Address channel	Connect to corresponding manager device, if populated, otherwise tie LOW
AWLOOP_S[1:0]	Input	Loopback signal	Connect to corresponding manager device, if populated, otherwise tie LOW
AWMPAM_S[10:0]	Input	MPAM signal: **AWMPAM[0]** MPAM_NS Security indicator, default = AWPROT[1]. **AWMPAM[9:1]** PARTID Partition identifier, default = `0x000`. **AWMPAM[10]** PMG Performance monitor group, default = `0b0`.	Connect to corresponding manager device, if populated, otherwise tie LOW
AWIDUNQ_S	Input	Unique ID indicator signal	Connect to corresponding manager device, if populated, otherwise tie LOW
AWNSAID_S[3:0]	Input	Non-secure Access Identifier signal	Connect to corresponding manager device, if populated, otherwise tie LOW This signal is not propagated by the CMN-700.
AWCMO_S[1:0]	Input	Write address channel CMO indicator	Connect to corresponding manager device, if populated, otherwise tie LOW
WREADY_S	Output	Write data ready	Connect to corresponding manager device, if populated
WVALID_S	Input	Write data valid	Connect to corresponding manager device, if populated, otherwise tie LOW
WDATA_S[n:0]	Input	Write data	Connect to corresponding manager device, if populated, otherwise tie LOW
WSTRB_S[d:0]	Input	Write byte lane strobes The value of `d` = (((n + 1) / 8) - 1).	Connect to corresponding manager device, if populated, otherwise tie LOW
WLAST_S	Input	Write data last transfer indication	Connect to corresponding manager device, if populated, otherwise tie LOW

Signal	Direction	Description	Connection information
WUSER_S[u:0]	Input	WUSER_S[0] is WDATACHK valid. If `META_DATA_EN` = 1, tie WUSER_S[u:1] to 0. The value of `u` depends on various parameters: • `u` = 0 if `META_DATA_EN` = 0. • `u` = 24 if `META_DATA_EN` = 1 and the AXI data bus is 512 bits wide. • `u` = 12 if `META_DATA_EN` = 1 and the AXI data bus is 128 bits wide.	Connect to corresponding manager device, if populated, otherwise tie LOW
WTRACE_S	Input	Trace signal	Connect to corresponding manager device, if populated, otherwise tie LOW
WPOISON_S[p:0]	Input	Poison signal The value of `p` = ceil(`DATA_WIDTH` / 64) - 1.	Connect to corresponding manager device, if populated, otherwise tie LOW
WDATACHK_S[d:0]	Input	Data check signal	Connect to corresponding manager device, if populated, otherwise tie LOW
BREADY_S	Input	Write response ready	Connect to corresponding manager device, if populated, otherwise tie LOW
BVALID_S	Output	Write response valid	Connect to corresponding manager 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	
BTRACE_S	Output	Trace signal	Connect to corresponding manager device, if populated
BLOOP_S[1:0]	Output.	Loopback signal	Connect to corresponding manager device, if populated
BIDUNQ_S	Output	Unique ID indicator signal	Connect to corresponding manager device, if populated
ARREADY_S	Output	Read address ready	Connect to corresponding manager device, if populated
ARVALID_S	Input	Read address valid	Connect to corresponding manager device, if populated, otherwise tie LOW
ARID_S[10:0]	Input	Read address ID	
ARADDR_S[n: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[n:0]	Input.	User-defined signal `n` = (`REQ_RSVDC_WIDTH` + `RSVDC_PBHA_WIDTH` (if enabled) + `RSVDC_STRONGNC_EN` (if enabled) -1)	
ARPROT_S[2:0]	Input	Read protection type	
ARQOS_S[3:0]	Input	Read QoS value	

Signal	Direction	Description	Connection information
ARSNOOP_S[3:0]	Input	Read transaction type	
ARDOMAIN_S[1:0]	Input	Read Shareability domain	
ARTRACE_S	Input	Trace signal	Connect to corresponding manager device, if populated, otherwise tie LOW
ARLOOP_S[1:0]	Input	Loopback signal	Connect to corresponding manager device, if populated, otherwise tie LOW
ARMPAM_S[10:0]	Input	MPAM signal: **ARMPAM[0]** MPAM_NS Security indicator, default = ARPROT[1]. **ARMPAM[9:1]** PARTID Partition identifier, default = `0x000`. **ARMPAM[10]** PMG Performance monitor group, default = `0b0`.	Connect to corresponding manager device, if populated, otherwise tie LOW
ARIDUNQ_S	Input	Unique ID indicator signal	Connect to corresponding manager device, if populated, otherwise tie LOW
ARCHUNKEN_S	Input	Chunk enable signal. If asserted, read data for this transaction can be returned out of order, in 128-bit chunks	Connect to corresponding manager device, if populated, otherwise tie LOW
ARNSAID_S[3:0]	Input	Non-secure Access Identifier signal	Connect to corresponding manager device, if populated, otherwise tie LOW This signal is not propagated by the CMN-700.
RREADY_S	Input	Read data ready	Connect to corresponding manager device, if populated, otherwise tie LOW
RVALID_S	Output	Read data valid	Connect to corresponding manager device, if populated
RID_S[10:0]	Output	Read data ID	Connect to corresponding manager device, if populated
RDATA_S[n:0]	Output	Read data	Connect to corresponding manager device, if populated
RRESP_S[2:0]	Output	Read data response	Connect to corresponding manager device, if populated
RLAST_S	Output	Read data last transfer indication	Connect to corresponding manager device, if populated

Signal	Direction	Description	Connection information
RUSER_S[x:0]	Output.	RUSER_S[0] is RDATACHK valid signal The value of `x` depends on various parameters: • `x` = 0 if `META_DATA_EN` = 0. • `x` = 24 if `META_DATA_EN` = 1 and the AXI data bus is 512 bits wide. • `x` = 12 if `META_DATA_EN` = 1 and the AXI data bus is 256 bits wide. • `x` = 6 if `META_DATA_EN` = 1 and the AXI data bus is 128 bits wide.	Connect to corresponding manager device, if populated
RTRACE_S	Output	Trace signal	Connect to corresponding manager device, if populated
RPOISON_S[p:0]	Output	Poison signal	Connect to corresponding manager device, if populated
RDATACHK_S[d:0]	Output	Data check signal	Connect to corresponding manager device, if populated
RLOOP_S[2:0]	Output	Loopback signal	Connect to corresponding manager device, if populated
RIDUNQ_S	Output	Unique ID indicator signal	Connect to corresponding manager device, if populated
RCHUNKV_S	Output	If asserted, RCHUNKNUM and RCHUNKSTRB are valid for this transfer	Connect to corresponding manager device, if populated
RCHUNKNUM_S	Output	Indicates the number of chunks being transferred. Chunks are numbered incrementally from zero, according to the data width and base address of the transaction.	Connect to corresponding manager device, if populated
RCHUNKSTRB_S	Output	Indicates which part of read data is valid for this transfer. Each bit corresponds to 128 bits of data. **RCHUNKSTRB[0]** Corresponds to RDATA[127:0]. **RCHUNKSTRB[1]** Corresponds to RDATA[255:128].	Connect to corresponding manager device, if populated
ACREADY_S	Input	Snoop address ready	Connect to corresponding manager device, if populated, otherwise tie LOW
ACVALID_S	Output	Snoop address valid	Connect to corresponding manager device, if populated
ACADDR_S[n:0]	Output	Snoop address	
ACSNOOP_S[3:0]	Output	Snoop transaction type	
ACPROT_S[2:0]	Output	Snoop protection type	
ACVMIDEXT[3:0]	Output	Snoop Address VMID Extension	
ACTRACE	Output	Snoop address trace	
CRREADY_S	Output	Snoop response ready	Connect to corresponding manager device, if populated
CRVALID_S	Input	Snoop response valid	Connect to corresponding manager device, if populated, otherwise tie LOW
CRRESP_S[4:0]	Input	Snoop response	
CRTRACE	Input	Snoop response trace	

WUSER_S[0] acts as a WDATACHK valid signal when `DATACHECK_EN` parameter is enabled:

- If WUSER_S[0] = 0, the RN-I or RN-D synthesizes the correct WDATACHK value before sending it on CHI write request.
- If WUSER_S[0] = 1, the RN-I or RN-D uses WDATACHK pin value to drive on CHI write request.

If the DATACHECK_EN parameter is disabled, theWUSER_S[0] input is ignored.

RUSER_S[0] acts as an RDATACHK valid signal. Since the RN-I or RN-D always drives the RDATACHK value, RUSER_S[0] is set to 1 when DATACHECK_EN parameter is enabled. If DATACHECK_EN parameter is disabled, RUSER_S[0] output is set to 0.

B.2.2 AXI/ACE-Lite manager interface signals

HN-I and SBSX have an AXI/ACE-Lite manager interface.

Signal definitions

Table B-2: AXI/ACE-Lite manager interface signals

Signal	Direction	Description	Connection information
ACLKEN_M	Input	AXI Manager bus clock enable signal	Connect to clock-enable logic
AWAKEUP_M	Output	Indicates that CMN-700 is starting an AXI transaction	Connect to corresponding subordinate device, if populated
AWREADY_M	Input	Write address ready	Connect to corresponding subordinate device, if populated, otherwise tie LOW.
AWVALID_M	Output	Write address valid.	Connect to corresponding subordinate device, if populated.
AWID_M[x:0]	Output	Write address ID For HN-I, x = 10. For HN-P, x = 17. For SBSX, x = 23. See B.2.3 Calculating the SBSX AxID signal widths on page 1279.	
AWADDR_M[n:0]	Output	Write address The value of n is configuration-dependent.	
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[n:0]	Output	User signal For HN-I, HN-D, HN-P,HN-V, HN-T and SBSX, n = (REQ_RSVDC_WIDTH + RSVDC_PBHA_WIDTH (if enabled) + RSVDC_STRONGNC_EN (if enabled) -1)	
AWPROT_M[2:0]	Output	Write protection type	
AWQOS_M[3:0]	Output	Write QoS value	
AWSNOOP_M[3:0]	Output	Shareable write transaction type	
AWDOMAIN_M[1:0]	Output	Write Shareability domain	
AWTRACE_M	Output	-	

Signal	Direction	Description	Connection information
AWMPAM_M[10:0]	Output	MPAM signal: **AWMPAM[0]** MPAM_NS Security indicator, default = AWPROT[1] **AWMPAM[9:1]** PARTID Partition identifier, default = `0x000` **AWMPAM[10]** PMG Performance monitor group, default = `0b0`	Connect to corresponding subordinate device, if populated
AWIDUNQ_M	Output	Unique ID indicator signal	Connect to corresponding subordinate device, if populated
AWNSAID_M[3:0]	Output	Non-secure Access Identifier signal	Connect to corresponding subordinate device, if populated This signal is not propagated by the CMN-700.
AWCMO_M[1:0]	Output	Type of CMO. This signal is only present on SBSX. It is not present on HN-I.	Connect to corresponding subordinate device, if populated
WREADY_M	Input	Write data ready	Connect to corresponding subordinate device, if populated, otherwise tie LOW
WVALID_M	Output	Write data valid	Connect to corresponding subordinate device, if populated
WDATA_M[n:0]	Output	Write data	Connect to corresponding subordinate device, if populated
WSTRB_M[n:0]	Output	Write byte lane strobe.	Connect to corresponding subordinate device, if populated
WLAST_M	Output	Write data last transfer indication	Connect to corresponding subordinate device, if populated
WUSER_M[u:0]	Output	WUSER_M[0] is WDATACHK valid signal The value of `u` depends on various parameters: • `u` = 0 if `META_DATA_EN` = 0. • `u` = 24 if `META_DATA_EN` = 1 and the AXI data bus is 512 bits wide. • `u` = 12 if `META_DATA_EN` = 1 and the AXI data bus is 256 bits wide. • `u` = 6 if `META_DATA_EN` = 1 and the AXI data bus is 128 bits wide.	Connect to corresponding subordinate device, if populated
WPOISON_M[p:0]	Output	Poison signal	Connect to corresponding manager device, if populated, otherwise tie LOW
WDATACHK_M[d:0]	Output	Data check signal	Connect to corresponding manager device, if populated, otherwise tie LOW
WTRACE_M	Output	Trace signal	Connect to corresponding manager device, if populated, otherwise tie LOW

[5] The value of `p` = (((n + 1) / 64) - 1).
[6] The value of `d` = (((n + 1) / 8) - 1).

Signal	Direction	Description	Connection information
BREADY_M	Output	Write response ready	Connect to corresponding subordinate device, if populated
BVALID_M	Input	Write response valid	Connect to corresponding subordinate device, if populated, otherwise tie LOW
BID_M[x:0]	Input	Write response ID	
BRESP_M[1:0]	Input	Write response	
BUSER_M[3:0]	Input	User signal	
BTRACE_M	Input	-	
BIDUNQ_M	Input	Unique ID indicator signal	Connect to corresponding subordinate device, if populated, otherwise tie LOW
BCOMP_M	Input	Write/CMO observable. This signal is only present on SBSX. It is not present on HN-I.	Connect to corresponding subordinate device, if populated, otherwise tie LOW
BPERSIST_M	Input	Data has been updated in persistent memory. This signal is only present on SBSX. It is not present on HN-I.	Connect to corresponding subordinate device, if populated, otherwise tie LOW
ARREADY_M	Input	Read address ready	Connect to corresponding subordinate device, if populated, otherwise tie LOW
ARVALID_M	Output	Read address valid	Connect to corresponding subordinate device, if populated
ARID_M[x:0]	Output	Read address ID	
ARADDR_M[n: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[n:0]	Output	User signal The value n = (`REQ_RSVDC_WIDTH` + `RSVDC_PBHA_WIDTH` (if enabled) + `RSVDC_STRONGNC_EN` (if enabled) - 1).	
ARPROT_M[2:0]	Output	Read protection type	
ARQOS_M[3:0]	Output	Read QoS value	
ARSNOOP_M[3:0]	Output	Shareable read transaction type	
ARDOMAIN_M[1:0]	Output	Read Shareability domain	
ARTRACE_M	Output	-	
ARMPAM_M[10:0]	Output	MPAM signal: **ARMPAM[0]** MPAM_NS Security indicator, default = ARPROT[1] **ARMPAM[9:1]** PARTID Partition identifier, default = `0x000` **ARMPAM[10]** PMG Performance monitor group, default = `0b0`	Connect to corresponding subordinate device, if populated
ARIDUNQ_M	Output	Unique ID indicator signal	Connect to corresponding subordinate device, if populated

Signal	Direction	Description	Connection information
ARNSAID_M[3:0]	Output	Non-secure Access Identifier signal	Connect to corresponding subordinate device, if populated This signal is not propagated by the CMN-700.
RREADY_M	Output	Read data ready	Connect to corresponding subordinate device, if populated
RVALID_M	Input	Read data valid	Connect to corresponding subordinate device, if populated, otherwise tie LOW
RID_M[x:0]	Input	Read data ID	Connect to corresponding subordinate device, if populated, otherwise tie LOW
RDATA_M[127:0]/ [255:0]	Input	Read data	Connect to corresponding subordinate device, if populated, otherwise tie LOW
RRESP_M[2:0]	Input	Read data response	Connect to corresponding subordinate device, if populated, otherwise tie LOW
RLAST_M	Input	Read data last transfer indication	Connect to corresponding subordinate device, if populated, otherwise tie LOW
RUSER_M[u:0]	Input	RUSER_M[0] is RDATACHK valid signal. If `META_DATA_EN` = 1, tie RUSER_M[u:1] to 0. Metadata_width + 1. Metadata_width + 2 if metadata is enabled	Connect to corresponding subordinate device, if populated, otherwise tie LOW
RPOISON_M[p:0]	Input	Poison signal	Connect to corresponding manager device, if populated, otherwise tie LOW
RDATACHK_M[d:0]	Input	Data check signal	Connect to corresponding manager device, if populated
RTRACE_M	Input	Trace signal	Connect to corresponding manager device, if populated, otherwise tie LOW
RIDUNQ_M	Input	Unique ID indicator signal	Connect to corresponding manager device, if populated, otherwise tie LOW

RUSER_M[0] acts as an RDATACHK valid signal when `DATACHECK_EN` parameter is enabled:

- If RUSER_M[0] = 0, the SBSX or HN-I synthesizes the correct RDATACHK value before sending it on CHI read data response.
- If RUSER_M[0] = 1, the SBSX or HN-I uses the RDATACHK pin value to drive CHI read data response.

If the `DATACHECK_EN` parameter is disabled, the RUSER_M[0] input is ignored.

WUSER_M[0] acts as a WDATACHK valid signal. Since the SBSX or HN-I always drives the WDATACHK value, WUSER_M[0] is set to 1 when `DATACHECK_EN` parameter is enabled. If the `DATACHECK_EN` parameter is not enabled, the WUSER_M[0] output is driven to 0.

B.2.3 Calculating the SBSX AxID signal widths

CMN-700 supports a width of 28, 24, or 9 bits for the AWID signals in SBSX.

The width of the AWID signals are determined by the following conditions:

1. Setting the CMO_ON_AW parameter in SBSX or the CHI_MTE_ENABLE global parameter to 1 or 0
2. The sum of the Tracker depth and the number of components being greater than 15. The values for Tracker depth and number of components is calculated by using the following equations:

   Tracker depth

   $\log_2$(NUM_DART)

   Number of Components

   $\log_2$(NUM(RND + RNF + RNI + CCG_HA + HNF + HNS))

The following shows how the above conditions determine the widths of the SBSX AWID signals:

AWID Width = 28 bits

1. CMO_ON_AW and POR_CHI_MTE_ENABLE are both set to 1
2. $\log_2$(NUM_DART) + $\log_2$(NUM(RND + RNF + RNI + CCG_HA + HNF + HNS)) > 15

AWID Width = 24 bits

1. CMO_ON_AW and POR_CHI_MTE_ENABLE are both set to 1
2. $\log_2$(NUM_DART) + $\log_2$(NUM(RND + RNF + RNI + CCG_HA + HNF + HNS)) < 15

AWID Width = 9 bits

1. CMO_ON_AW and POR_CHI_MTE_ENABLE are both set to 0

The following table shows an example of the number of active bits in a AWID Width of 28, that is calculated using a max configuration where both the previous conditions are met, the NUM_DART parameter is 128, and the total number of components is 506.

Table B-3: Sizing of the SBSX AWID: 28 bits

AWID Width	Field	Field width [Top to bottom = LSB to MSB fields]	Max Values
28	Reserved	[27:26]	2
	IS2PART	[25:25]	1
	Logical ReturnNID	[24:16]	9
	PGroupID	[15:8]	8
	Tracker ID	[7:0]	8

The following table shows an example of the number of active bits in a AWID Width of 24, that is calculated using a configuration where only the first condition is met, the NUM_DART parameter is 128, and the total number of components is 248.

Table B-4: Sizing of the SBSX AWID: 24 bits

AWID Width	Field	Field width [Top to bottom = LSB to MSB fields]	Max Values
24	Reserved	[23:23]	1
	IS2PART	[22:22]	1
	Logical ReturnNID	[21:15]	7
	PGroupID	[14:7]	8
	Tracker ID	[6:0]	7

The following table shows an example of the number of active bits in a AWID Width of 9, that is calculated using a configuration where the first condition is not met.

Table B-5: Sizing of the SBSX AWID

AWID Width	Field	Field width [Top to bottom = LSB to MSB fields]	Max Values
9	Tracker ID	[8:0]	9

B.2.4 HN-I and HN-P AxID signal properties and encodings

The size and encodings of the AxID signals on the AXI or ACE-Lite interface are different for HN-I and HN-P.

The term *peer-to-peer* refers to requests from an RN-I or RN-D that is connected to a PCIe-RC to an HN-I that is connected to a PCIe-RC or an HN-P. The pcie_mstr_present bit in the por_rn{i,d}_cfg_ctl register indicates whether an RN-I or RN-D is connected to a PCIe-RC. Also, dis_awid_to_hni_hnp_cxra bit in the por_rn{i,d}_cfg_ctl register, must remain set to 0 to enable passing the AxID signal to HN-I and HN-P. The physical_mem_en attribute can be enabled in the SAM for an address range where Normal memory ordering is required. For example, prefetchable MMIO or non-prefetchable MMIO space which is mapped as Normal memory.

The following table shows the AxID properties and encodings for the HN-I and HN-P.

Table B-6: HN-I and HN-P AxID signal encodings

Device type	AxID width	Downstream memory type	Request source	AW or AR	Encoding
HN-I	11	physical_mem_en = 0	Any	AWID or ARID	**AxID[3:0]** HN-I SAM order region encoding **AxID[4]** Reserved **AxID[5]** PCIe write coloring **AxID[7:6]** HN-I SAM address region encoding **AxID[10:8]** Reserved
		physical_mem_en = 1	Any	AWID or ARID	**AxID[4:0]** UniqId[4:0] **AxID[5]** PCIe write coloring **AxID[7:6]** Address region encoding **AxID[10:8]** UniqId[7:5]
HN-P	18	physical_mem_en = 0	Non peer-to-peer	AWID or ARID	**AxID[10:0]** Same as HN-I Device memory **AxID[17:11]** Reserved

Device type	AxID width	Downstream memory type	Request source	AW or AR	Encoding
		physical_mem_en = 1	Non peer-to-peer	AWID or ARID	**AxID[10:0]** Same as HN-I Normal memory **AxID[17:11]** Reserved
		N/A (see note)	Peer-to-peer	AWID	**AWID[5:0]** Hash of PCIe RN-I AWID **AWID[12:6]** RN-I Logical ID **AWID[13]** RN-I vs. RN-D indicator (1 if RN-I, 0 if RN-D) **AWID[14]** PCIe RN-I/RN-D indicator (1 if PCIe RN-I or RN-D) **AWID[16:15]** Chip ID (Ranges from 0 to 3) **AWID[17]** Peer-to-peer coloring
		N/A (see note)	Peer-to-peer	ARID	**ARID[`HNP_RD_NUM_AXI_REQS_PARAM_LOG2`-1:0]** HN-P read RDT entry **ARID[16: `HNP_RD_NUM_AXI_REQS_PARAM_LOG2`]** Reserved **AWID[17]** Peer-to-peer coloring

Note

Peer-to-Peer traffic in HN-P does not generate AXIDs or order transactions based on downstream memory type.

B.2.5 A4S signals

The A4S interface signals are listed in the following tables.

Signal definitions

Table B-7: A4S Transmit and Receive signals

Signal	Direction	Description	Connection information
TXA4STREADY	Input	TXA4STREADY indicates that the subordinate can accept a transfer in the current cycle	Connect from RXA4STREADY of the A4S subordinate, if populated, otherwise tie LOW
TXA4STVALID	Output	TXA4STVALID indicates that the manager is driving a valid transfer. A transfer takes place when both TXA4STVALID and TXA4STREADY are asserted.	Connect to RXA4STVALID of the A4S subordinate, if populated
TXA4STDEST[9:0]	Output	`0b00000000`	TXA4STDEST is always zero
TXA4STID[9:0]	Output	TXA4STID is the data stream identifier that indicates different streams of data	Connect to RXA4STID of the A4S subordinate, if populated
TXA4STDATA[63:0]	Output	TXA4STDATA is the primary payload that is used to provide the data that is passing across the interface	Connect to RXA4STDATA of the A4S subordinate, if populated

Signal	Direction	Description	Connection information
TXA4STSTRB[7:0]	Output	TXA4STSTRB is the byte qualifier that indicates whether the content of the associated byte of TXA4STDATA is processed as a data byte or a position byte	Connect to RXA4STSTRB of the A4S subordinate, if populated
TXA4STKEEP[7:0]	Output	TXA4STKEEP is the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as part of the data stream. Associated bytes that have the TKEEP byte qualifier deasserted are null bytes and can be removed from the data stream.	Connect to RXA4STKEEP of the A4S subordinate, if populated
TXA4STLAST	Output	TXA4STLAST indicates the boundary of a packet	Connect to RXA4STLAST of the A4S subordinate, if populated
RXA4STREADY	Output	RXA4STREADY indicates that the subordinate can accept a transfer in the current cycle	Connect to TXA4STREADY of the A4S manager, if populated
RXA4STVALID	Input	RXA4STVALID indicates that the manager is driving a valid transfer. A transfer takes place when both RXA4STVALID and RXA4STREADY are asserted.	Connect from TXA4STVALID of the A4S manager, if populated, otherwise tie LOW
RXA4STDEST[9:0]	Input	RXA4STDEST provides routing information for the data stream	Connect from TXA4STDEST of the A4S manager, if populated, otherwise tie LOW
RXA4STID[9:0]	Input	RXA4STID is the data stream identifier that indicates different streams of data	Connect from TXA4STID of the A4S manager, if populated, otherwise tie LOW
RXA4STRI[5:0]	Input	RXA4STRI is the chip to chip routing information that indicates RA ID of the other chip	Connect from TXA4STRI of the A4S manager, if populated, otherwise tie LOW
RXA4STDATA[63:0]	Input	RXA4STDATA is the primary payload that is used to provide the data that is passing across the interface	Connect from TXA4STDATA of the A4S manager, if populated, otherwise tie LOW
RXA4STSTRB[7:0]	Input	RXA4STSTRB is the byte qualifier that indicates whether the content of the associated byte of RXA4STDATA is processed as a data byte or a position byte	Connect from TXA4STSTRB of the A4S manager, if populated, otherwise tie LOW
RXA4STKEEP[7:0]	Input	RXA4STKEEP is the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as part of the data stream. Associated bytes that have the TKEEP byte qualifier deasserted are null bytes and are removed from the data stream.	Connect from TXA4STKEEP of the A4S manager, if populated, otherwise tie LOW
RXA4STLAST	Input	RXA4STLAST indicates the boundary of a packet	Connect from TXA4STLAST of the A4S manager, if populated, otherwise tie LOW

B.2.6 AXU interface signals

The following table lists the AXU interface signals.

Signal definitions

Table B-8: AXU interface signals

Signal	Direction	Description	Connection information
ACLKENU	Input	Global clock signal. Synchronous signals are sampled on the rising edge of the global clock.	Connect to clock enable logic.
AWAKEUPU	Output	Manager Wakeup_Signals indicates that activity is initiated on the write or read address channels.	Connect to corresponding subordinate device, if populated.
AWREADYU	Input	Subordinate indicates that a transfer on the write address channel can be accepted.	Connect to corresponding subordinate device, if populated, otherwise tie LOW.
AWADDRU[23:0]	Output	The address of the write transaction.	Connect to corresponding subordinate device, if populated.
AWLENU[7:0]	Output	The number of data transfers in a write transaction.	
AWSIZEU[2:0]	Output	The number of bytes in each data transfer in a write transaction.	
AWBURSTU[1:0]	Output	Burst type, indicates how address changes between each transfer in a write transaction.	
AWLOCKU	Output	Provides information about the atomic characteristics of a write transaction.	
AWCACHEU[3:0]	Output	Indicates how a write transaction is required to progress through a system.	
AWPROTU[2:0]	Output	Protection attributes of a write transaction: privilege, security level, and access type.	
WREADYU	Input	Indicates that a transfer on the write data channel can be accepted.	Connect to corresponding subordinate device, if populated, otherwise tie LOW.
WVALIDU	Output	Indicates that the write data channel signals are valid.	Connect to corresponding subordinate device, if populated.
WDATAU[63:0]	Output	Write data.	
WSTRBU[7:0]	Output	Write strobes, indicate which byte lanes hold valid data.	
WLASTU	Output	Indicates whether this is the last data transfer in a write transaction.	
ARREADYU	Input	Subordinate indicates that a transfer on the read address channel can be accepted.	Connect to corresponding subordinate device, if populated, otherwise tie LOW.
ARVALIDU	Output	Manager indicates that the read address channel signals are valid.	Connect to corresponding subordinate device, if populated.
ARADDRU[23:0]	Output	The address of the read transaction.	
ARLENU[7:0]	Output	The number of data transfers in a read transaction.	
ARSIZEU[2:0]	Output	The number of bytes in each data transfer in a read transaction.	
ARBURSTU[1:0]	Output	Burst type, indicates how address changes between each transfer in a read transaction.	
ARLOCKU	Output	Provides information about the atomic characteristics of a read transaction.	
ARCACHEU[3:0]	Output	Indicates how a read transaction is required to progress through a system.	

Signal	Direction	Description	Connection information
ARPROTU[2:0]	Output	Protection attributes of a read transaction: privilege, security level, and access type.	
BREADYU	Output	Manager indicates that a transfer on the write response channel can be accepted.	Connect to corresponding subordinate device, if populated.
BVALIDU	Input	Subordinate indicates that the write response channel signals are valid.	Connect to corresponding subordinate device, if populated, otherwise tie LOW.
BRESPU[1:0]	Input	Subordinate Write response, indicates the status of a write transaction.	
RREADYU	Output	Manager indicates that a transfer on the read data channel can be accepted.	Connect to corresponding subordinate device, if populated.
RVALIDU	Input	Subordinate indicates that the read data channel signals are valid.	Connect to corresponding subordinate device, if populated, otherwise tie LOW.
RDATAU[63:0]	Input	Subordinate Read data.	
RRESPU[1:0]	Input	Subordinate Read response, indicates the status of a read transfer.	
RLASTU	Input	Subordinate indicates whether this is the last data transfer in a read transaction.	

B.3 APB interface signals

HN-D nodes have an APB interface to support the connection of an external APB manager device.

Each CML Gateway (CCG) block has an APB port for CXL controller IP to access CXL defined registers implemented in them.


Note

All signal names in this section are only a root name indicated as RootName.
CMN-700 interfaces use RootName within a more fully specified signal name as follows:
CMN-700 interface signal name == RootName_NID#, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-9: APB signals

Signal	Direction	Description	Connection information
PADDR[31:0]	Input	Address that is associated with the APB transaction	Connect to corresponding ports on external APB manager device **Note:** CCG APB Port has 14-bit address PADDR[13:0]
PPROT[2:0]	Input	Protection type of the transaction	
PSEL	Input	Indicates that the subordinate device is selected and that a data transfer is required	
PENABLE	Input	Enable. Indicates the second and subsequent cycles of an APB transfer.	
PWRITE	Input	Indicates that the access is a write when HIGH. Indicates that the access is a read when LOW.	
PWDATA[31:0]	Input	Write data	

Signal	Direction	Description	Connection information
PSTRB[3:0]	Input	Write strobes	
PREADY	Output	Ready	
PRDATA[31:0]	Output	Read data	
PSLVERR	Output	Indicates a transfer failure	

B.4 ATPG interface signals

The following contains information on ATPG interface signal information.

Signal definitions

Table B-10: DFT interface signals

Signal	Direction	Description	Connection information
DFTCLKBYPASS	Input	Select the SLC RAM clock to follow the CMN-700 input clock, as applicable for each clock region	Tie LOW if unused
DFTCLKDISABLE[3:0]	Input	Disable clock regions during scan shift	
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	
DFTCGEN	Input	Scan shift enable. Forces on the clock grids during scan shift.	
DFTSCANMODE	Input	During functional mode, the HN-F SLC 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 enable holds MCPs to be used on the RAM set address and write data inputs in the implementation flow and during static timing analysis.	

B.4.1 Block-level ATPG signals

CMN-700 supports DFT using the ATPG methodology. The design contains various signals that are used to carry out ATPG testing.

The following table lists the ATPG signals at the HN-F block-level.

Table B-11: HN-F block-level cmn_hnf ATPG signals

Signal	Direction	Description
DFTCLKBYPASS	Input	Bypasses stretch RAM clock for `STRETCH_L3RAMCLK configuration parameter. Functional mode = 0b0.
DFTRSTDISABLE	Input	Disables internal synchronized reset during scan shift. Functional mode = 0b0.
DFTCLKDISABLE	Input	Disables clock regions during test to save power. Functional mode = 0b0.
DFTCGEN	Input	Overrides clock gate shift. Functional mode = 0b0.
DFTRAMHOLD	Input	Blocks chip select to RAMs to preserve state. Functional mode = 0b0.
DFTMCPHOLD	Input	Limits number of multicycle path toggles during ATPG delay tests. Functional mode = 0b0.
DFTSCANMODE	Input	Prevents potential RAM input hold violations during ATPG. Functional mode = 0b0.
clk_por	Input	Functional clock.
nSKYRESET	Input	Functional reset. Active-LOW.
nMBISTRESET	Input	MBIST mode entry reset. Functional mode = 0b1. Active-LOW.

The following table lists the ATPG signals at the HN-D block-level (por_hnd), at the HN-I block-level (por_hni), and at the SBSX block-level (por_sbsx).

Table B-12: HN-D/HN-I/SBSX block-level por_hnd/por_hni/por_sbsx ATPG signals

Signal	Direction	Description
DFTRSTDISABLE	Input	Disables internal synchronized reset during scan shift. Functional mode = 0b0.
DFTCLKDISABLE	Input	Disables clock regions during test to save power. Functional mode = 0b0.
DFTCGEN	Input	Overrides clock gate shift. Functional mode = 0b0.
DFTCLKBYPASS	Input	Bypasses stretch RAM clock for `STRETCH_L3RAMCLK configuration parameter. Functional mode = 0b0.
clk_por	Input	Functional clock.
nCHIRESET	Input	Functional reset. Active-LOW.

The following table lists the ATPG signals at the RN-I block-level (por_rni) and at the RN-D block-level (por_rnd).

Table B-13: RN-I/RN-D block-level por_rni/por_rnd ATPG signals

Signal	Direction	Description
DFTCLKBYPASS	Input	Bypasses stretch RAM clock for `STRETCH_L3RAMCLK configuration parameter. Functional mode = 0b0.
DFTRSTDISABLE	Input	Disables internal synchronized reset during scan shift. Functional mode = 0b0.
DFTCLKDISABLE	Input	Disables clock gate regions during test to save power. Functional mode = 0b0.
DFTCGEN	Input	Overrides clock gate shift. Functional mode = 0b0.
DFTRAMHOLD	Input	Blocks chip select to RAMs to preserve state. Functional mode = 0b0.
DFTSCANMODE	Input	Prevents violation of the HN-F RAM input multicycle hold paths on data and address RAM inputs. Signal is only present on MTSX ATPG interface.
clk_por	Input	Functional clock.
nCHIRESET	Input	Functional reset. Active-LOW.
nMBISTRESET	Input	MBIST mode entry reset. Functional mode = 0b1. Active-LOW.

The following table lists the ATPG signals as internal pins for SMXP blocks named por_smxp_*. Arm recommends that you use the drivers of these internal pins for any DFT purposes during synthesis.

Table B-14: SMXP block-level por_smxp_* ATPG signals

Signal	Direction	Description
u_mxp_misc.mxp_dftrstdisable	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
u_mxp_misc.mxp_dftclkdisable	Input	Disables clock regions during test to save power. Functional mode = `0b0`.
u_mxp_misc.mxp_dftcgen	Input	Overrides clock gate shift. Functional mode = `0b0`.
clk_por	Input	Functional clock.
u_mxp_misc.mxp_nporreset	Input	Functional reset. Active-LOW.

The following table lists the ATPG signals as internal nets for MCS blocks named por_mcsx and por_mcsy. Arm recommends that you use the drivers of these internal pins for any DFT purposes during synthesis.

Table B-15: MCS block-level por_mcsx/por_mcsy ATPG signals

Signal	Direction	Description
mcs_dftrstdisable	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
mcs_dftcgen	Input	Overrides clock gate shift. Functional mode = `0b0`.
clk_por	Input	Functional clock.
mcs_nporreset	Input	Functional reset. Active-LOW.

The following table lists the ATPG signals as internal nets for DCS blocks named por_dcs_*. Arm® recommends that you use the drivers of these internal pins for any DFT purposes during synthesis.

Table B-16: DCS block-level por_dcs_* ATPG signals

Signal	Direction	Description
dcs_dftrstdisable	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
dcs_dftcgen	Input	Overrides clock gate shift. Functional mode = `0b0`.
clk_por	Input	Functional clock.
dcs_nporreset	Input	Functional reset. Active-LOW.

The following table lists the ATPG signals as internal nets for CCS blocks named por_ccs_*. Arm® recommends that you use the drivers of these internal pins for any DFT purposes during synthesis.

Table B-17: CCS block-level por_ccs_* ATPG signals

Signal	Direction	Description
ccs_dftrstdisable	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
ccs_dftcgen	Input	Overrides clock gate shift. Functional mode = `0b0`.
clk_por	Input	Functional clock.
ccs_nporreset	Input	Functional reset. Active-LOW.

The following table lists the ATPG signals as internal nets for CAL blocks named por_cal{2,4}_*. Arm® recommends that you use the drivers of these internal pins for any DFT purposes during synthesis.

Table B-18: CAL block-level por_cal{2,4}_* ATPG signals

Signal	Direction	Description
cal_dftrstdisable	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
cal_dftcgen	Input	Overrides clock gate shift. Functional mode = `0b0`.
clk_por	Input	Functional clock.
cal_nporreset	Input	Functional reset. Active-LOW.

The following table lists the ATPG signals at the PDB block-level (pdb_rnf) and at the CDB block-level (cdb_rnf/cdb_snf).

Table B-19: PDB/CDB block-level pdb_rnf/cdb_rnf/cdb_snf ATPG signals

Signal	Direction	Description
DFTRSTDISABLE	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
DFTCGEN	Input	Overrides clock gate shift. Functional mode = `0b0`.
CLK_DEV, CLK_ICN	Input	Functional clocks.
RESETN_DEV, RESETN_ICN	Input	Functional resets. Active-LOW.

The following table lists the ATPG signals at the ADB block-level (adb_*).

Table B-20: ADB block-level adb_* ATPG signals

Signal	Direction	Description
DFTRSTDISABLE	Input	Disables internal synchronized reset during scan shift. Functional mode = `0b0`.
DFTCGEN	Input	Overrides clock gate shift. Functional mode = `0b0`.
CLK_S, CLK_M	Input	Functional clocks.
RESETN_S, RESETN_M	Input	Functional resets. Active-LOW.

B.5 CHI interface signals

CMN-700 uses channels that form an inbound and outbound CHI interface for each device using signals that form each channel in a specific interface.

The *AMBA® 5 CHI Architecture Specification* defines four channels:

- *Request* (REQ)
- *Response* (RSP)
- *Snoop* (SNP)
- *Data* (DAT)


Note

All signal names in this section are only a root name, RootName. CMN-700 interfaces use RootName within a more fully specified signal name as follows:

- CMN-700 interface signal name == RootName_NID#, where # is the node ID corresponding to the specific interface.

B.5.1 Per-device interface definition

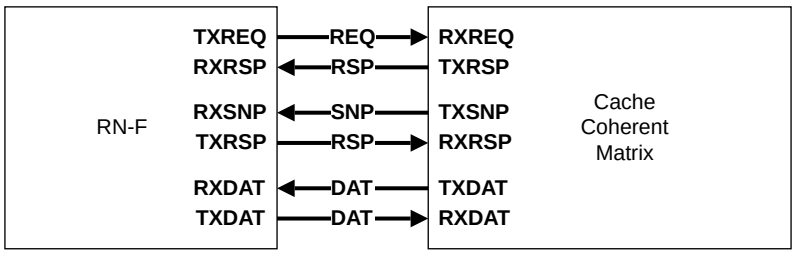
Each CHI device included in a CMN-700 system has distinct functionality, and the requirements and configuration of its respective CHI interfaces differ.

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.

Figure B-1: External RN-F interface

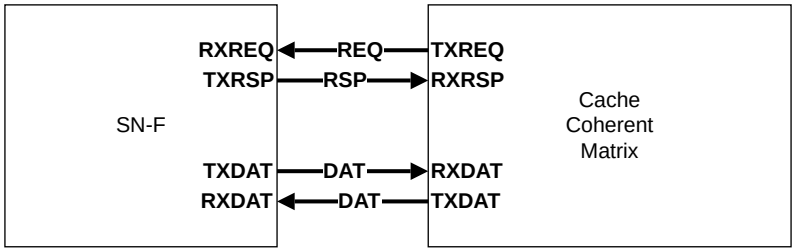


It also has two data channels, one in each direction, for data transfers. CMN-700 receives request messages from the RN-F and sends responses to it. In addition, CMN-700 sends snoop messages to the RN-F and receives snoop response messages.

External SN-F interface

The SN-F interface consists of a request channel and a response channel as the following figure shows.

Figure B-2: External SN-F interface



It also has two data channels, one in each direction, for data transfers. The SN-F receives request messages from CMN-700 and returns response messages.

B.5.2 Per-channel interface signals

For communication between devices, each channel includes a *Transmit* (TX) and a *Receive* (RX) port, with various interface signals traveling from TX to RX.

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.

Signal definitions

Table B-21: Per-channel interface signals

Signal	Direction	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[n:0][7]	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
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[n:0][7]	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
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[n:0][7]	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
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[n:0][7]	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
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

[7] The value of n is configuration-dependent.

Signal	Direction	Description	Connection information
RXREQFLIT[n:0][7]	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
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[n:0][7]	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
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[n:0][7]	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
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[n:0][7]	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

B.5.3 Non-channel-specific interface signals

Every transmit and receive link layer interface includes extra signals that exist only at the interface level and are not channel specific.

Signal definitions

Table B-22: Non-channel-specific interface signals

Signal	Direction	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
TXLINKACTIVEREQ	Output	Transmit channel LinkActive request to adjacent receiver device	Connect to RXLINKACTIVEREQ of the corresponding CHI device, if populated

Signal	Direction	Description	Connection information
TXLINKACTIVEACK	Input	Transmit channel LinkActive acknowledgment from adjacent receiver device	Connect to RXLINKACTIVEACK of the corresponding CHI device, if populated, otherwise tie LOW
RXSACTIVE	Input	Indication from the adjacent CHI device that it has one or more outstanding protocol-layer transactions. RXSACTIVE must remain asserted throughout the lifetime of the transaction.	Connect to TXSACTIVE of the corresponding CHI device
TXSACTIVE	Output	Indication to the adjacent CHI device that CMN-700 has one or more outstanding protocol-layer transactions. TXSACTIVE remains asserted throughout the lifetime of the transaction.	Connect to RXSACTIVE of the corresponding CHI device
SYSCOREQ	Input	Request to enter CHI coherence domain when asserted and to exit the CHI coherence domain when deasserted. SYSCOREQ and SYSCOACK implement a four-phase handshake protocol.	Connect to SYSCOREQ of corresponding CHI device, if populated, otherwise tie LOW
SYSCOACK	Output	Acknowledge CHI coherence domain entry/exit request	Connect to SYSCOACK of corresponding CHI device, if populated

B.5.4 RSVDC signal description

The following describes subfields of RSVDC fields on CHI channels and corresponding *USER fields on ACE-Lite channels, where applicable.

Table B-23: CHI REQ.RSVDC/AWUSER/ARUSER

Position	Field	HN-P	HN-I*	RN-I/RN-D	CHI
MSB	Last	Y	N	N	N
LSB	RSVDC	Y	Y	Y	Y

Table B-24: CHI DAT.RSVDC/WUSER/RUSER

Position	Field	HN-P	HN-I*	RN-I/RN-D	CHI
MSB	METADATA	Y	Y	Y	Y
	RSVDC	N	N	N	Y
LSB	DataCheck valid	Y	Y	Y	N

The RUSER and WUSER signal widths increase on ACE-Lite manager and subordinate interfaces when RSVDC_METADATA_MODE_EN is set to 1. However they are not used to propagate DAT RSVDC values.

Table B-25: DAT.RSVDC subfields description

Signal	Type	Description	Connection information
BASE[BASE_WIDTH-1:0]	Input/output	Base	BASE_WIDTH is 4
METADATA[METADATA_WIDTH-1:0]	Input/output	Metadata	RSVDC_METADATA_WIDTH=12 for DAT.RSVDC. This field does not exist if RSVDC_METADATA_MODE_EN = 0


Note

The preceding subfields are fully packed in the same order as represented in the table to create DAT RSVDC field where BASE subfield represents LSB side.

Table B-26: RSVDC value override

Signal	Type	Description	Connection Information
REQ_RSVDC_OVRD_VAL [POR_DEV_REQ_RSVDC_WIDTH_PARAM-1:0]	Input	Strap to override REQ Flit RSVDC value at each CCG when configured for non-SMP connection	Drive from SoC or Tie LOW if unused
DAT_RSVDC_OVRD_VAL [POR_DEV_DAT_RSVDC_WIDTH_PARAM-1:0]	Input	Strap to override DAT Flit RSVDC value at each CCG when configured for non-SMP connection	Drive from SoC or Tie LOW if unused

B.6 Clock and reset signals

CMN-700 has global clock and reset signals for controlling the clock and reset functionality.

Depending on your CMN-700 clock domain configuration, the interconnect might have a single global clock signal or multiple global clock signals.

Signal definitions

Table B-27: CMN-700 clock and reset signals

Signal	Direction	Description	Connection information
GCLK0	Input	Primary CMN-700 clock input. This clock signal is always present.	Connect to global clock for CMN-700.
GCLK1	Input	Primary CMN-700 clock input. This clock is only present if you divide the mesh into four asynchronous clock domains.	Connect to global clock for CMN-700.
GCLK2	Input	Primary CMN-700 clock input. This clock is only present if you divide the mesh into four asynchronous clock domains.	Connect to global clock for CMN-700.
GCLK3	Input	Primary CMN-700 clock input. This clock is only present if you divide the mesh into four asynchronous clock domains.	Connect to global clock for CMN-700.
nSRESET	Input	CMN-700 reset, active-LOW	Connect to global reset for CMN-700.

B.7 Clock management signals

The following contains information on clock management Q-Channel signals in the CMN-700.

Signal definitions

Table B-28: Clock management Q-Channel signals

Signal	Direction	Description	Connection information
QACTIVE_CLKCTL	Output	Indication that CMN-700 is active, and that the *External Clock Controller* (ExtCC) must not make a request for CMN-700 to prepare to stop the clocks.	Connect to external clock controller
QREQn_CLKCTL	Input	Request from the ExtCC for the CMN-700 to prepare to stop the clocks. Active-LOW.	Connect to external clock controller or tie HIGH if unused
QACCEPTn_CLKCTL	Output	Positive acknowledgment after receiving QREQn assertion indicating that CMN-700 has completed preparation to stop the clocks and that the ExtCC can stop the clocks. Active-LOW.	Connect to external clock controller
QDENY_CLKCTL	Output	Negative acknowledgment after receiving QREQn assertion indicating that CMN-700 has refused the request from the ExtCC to prepare to stop the clocks	

B.8 CML clock management signals

The following contains information on clock management Q-Channel signals when using CML.

Signal definitions

Table B-29: Clock management Q-Channel signals

Signal	Direction	Description	Connection information
QACTIVE_CXSCLKCTL	Output	Indication that the CCL side of CMN-700 is active, and that the *External Clock Controller* (ExtCC) must not make a request for CMN-700 and corresponding CCG device to prepare to stop the clock CLK_CXS.	OR with QACTIVE_CXSCLKCTL (CXSDB) and connect to external clock controller
QREQn_CXSCLKCTL	Input	Request from the ExtCC for the CMN-700 to prepare to stop the clock CLK_CXS. Active-LOW.	Connect to external clock controller or tie HIGH if unused
QACCEPTn_CXSCLKCTL	Output	Positive acknowledgment after receiving QREQn assertion indicating that CMN-700 has completed preparation to stop the clock CLK_CXS and that the ExtCC can stop the clock CLK_CXS. Active-LOW.	Connect to external clock controller
QDENY_CXSCLKCTL	Output	Negative acknowledgment after receiving QREQn assertion indicating that CMN-700 has refused the request from the ExtCC to prepare to stop the clock CLK_CXS	

B.9 Configuration input signals

CMN-700 has configuration input signals for driving specific configuration values for the interconnect.

All of these signals must be stable at least ten cycles before deassertion of reset. These signals must remain stable throughout the operation of CMN-700, until a following reset assertion or power down, if any.

Signal definitions

Table B-30: Configuration input signals

Signal	Direction	Description	Connection information
CFGM_PERIPHBASE[47:28]	Input	Base address [47:28] of the CMN-700 configuration register space	Tie as required for system memory map
GICD_DESTID[15:0]	Input	A4S Logical ID of GICD connection	Tie as required for CML GICD communication
DMC_PERIPHBASE[51:28]	Input	Base address of the DMC configuration space	Tie as required for system memory map
DSU_PERIPHBASE[51:28]	Input	Base address of the DSU configuration space LSB corresponds to the rounded up two's power of total DSU space	Tie as required for system memory map

B.10 CXS interface signals

The following tables describe CMN-700 CXS interface signals.

Each CML Gateway (CCG) block have a CXS interface to communicate with external multi chip controller IP.

All signal names in this section are only a root name indicated as RootName.
CMN-700 interfaces use RootName within a more fully specified signal name as follows:
CMN-700 interface signal name == RootName_NID#, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-31: Transmit and Recieve signals

Signal	Direction	Description	Connection information
CXSTXDATA[511:0]	Output	Transmit channel data flit	Connect to CXSRXDATA of the corresponding PCIe IP
CXSRXPRCLTYPE[2:0]	Output	Transmit channel protocol type indicator	Connect to CXSRXPRCLTYPE of the corresponding PCIe IP
CXSRXLAST	Output	Transmit channel flit insertion indicator	Connect to CXSRXLAST of the corresponding PCIe IP

Signal	Direction	Description	Connection information
CXSTXCNTL[17:0]	Output	Transmit channel control information	Connect to CXSRXCNTL of the corresponding PCIe IP
CXSTXVALID	Output	Transmit channel data flit valid	Connect to CXSRXVALID of the corresponding PCIe IP
CXSTXCRDGNT	Input	Transmit channel link layer credit grant	Connect to CXSRXCRDGNT of the corresponding PCIe IP
CXSTXCRDRTN	Output	Transmit channel link layer credit return	Connect to CXSRXCRDRTN of the corresponding PCIe IP
CXSTXACTIVEREQ	Output	Transmit channel link activation/deactivation request	Connect to CXSRXACTIVEREQ of the corresponding PCIe IP
CXSTXACTIVEACK	Input	Transmit channel link activation/deactivation acknowledge	Connect to CXSRXACTIVEACK of the corresponding PCIe IP
CXSTXDEACTHINT	Input	Transmit channel hint for link deactivation	Connect to CXSRXDEACTHINT of the corresponding PCIe IP
CXSRXDATA[511:0]	Input	Receive channel data flit	Connect to CXSTXDATA of the corresponding PCIe IP
CXSRXPRCLTYPE[2:0]	Output	Transmit channel protocol type indicator	Connect to CXSRXPRCLTYPE of the corresponding PCIe IP
CXSRXLAST	Output	Transmit channel flit insertion indicator	Connect to CXSRXLAST of the corresponding PCIe IP
CXSRXCNTL[17:0]	Input	Receive channel control information	Connect to CXSTXCNTL of the corresponding PCIe IP
CXSRXVALID	Input	Receive channel data flit valid	Connect to CXSTXVALID of the corresponding PCIe IP
CXSRXCRDGNT	Output	Receive channel link layer credit grant	Connect to CXSTXCRDGNT of the corresponding PCIe IP
CXSRXCRDRTN	Input	Receive channel link layer credit return	Connect to CXSTXCRDRTN of the corresponding PCIe IP
CXSRXACTIVEREQ	Input	Receive channel link activation/deactivation request	Connect to CXSTXACTIVEREQ of the corresponding PCIe IP
CXSRXACTIVEACK	Output	Receive channel link activation/deactivation acknowledge	Connect to CXSTXACTIVEACK of the corresponding PCIe IP
CXSRXDEACTHINT	Output	Receive channel hint for link deactivation	Connect to CXSTXDEACTHINT of the corresponding PCIe IP

B.11 Debug, trace, and PMU interface signals

Signals that aid debugging are included in CMN-700.


Note

All signal names in this section are only a root name indicated as RootName. CMN-700 interfaces use RootName within a more fully specified signal name as follows:

CMN-700 interface signal name == RootName_NID#, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-32: Debug, trace, and PMU interface signals

Signal	Direction	Description	Connection information
ATCLKEN_NID<x>	Input	ATB clock enable, where <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
ATREADY_NID<x>	Input	ATB device ready: **0** Not ready **1** Ready <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
AFVALID_NID<x>	Input	FIFO flush request, where <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
ATDATA[31:0]_NID<x>	Output	ATB data bus, where <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
ATVALID_NID<x>	Output	ATB valid data: **0** No valid data **1** Valid data <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
ATBYTES[1:0]_NID<x>	Output	CoreSight ATB device data size: **0b00** 1 byte **0b01** 2 bytes **0b10** 3 bytes **0b11** 4 bytes <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
AFREADY_NID<x>	Output	FIFO flush acknowledge: **0** FIFO flush not complete **1** FIFO flush complete <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
ATID[6:0]_NID<x>	Output	ATB trace source identification, where <x> is the NodeID number for that HN-D/DTC or HN-T/DTC	-
DBGWATCHTRIGREQ_NID<x>	Output	Trigger output from DEM indicating assertion of a DT event. DBGWATCHTRIGREQ is asynchronous-safe, and operates in a four-phase handshake with DBGWATCHTRIGACK. <x> is the NodeID number for that HN-D/DTC or HN-T/DTC.	Connect to external debug and trace control logic
DBGWATCHTRIGACK_NID<x>	Input	External acknowledgment of receipt of DBGWATCHTRIGREQ. DBGWATCHTRIGACK must be asynchronous-safe, and operates in a four-phase handshake with DBGWATCHTRIGREQ. <x> is the NodeID number for that HN-D/DTC or HN-T/DTC.	Connect to external debug and trace control logic, or tie LOW if DBGWATCHTRIGREQ is unused

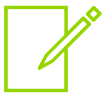
Signal	Direction	Description	Connection information
PMUSNAPSHOTREQ	Input	External request that the live PMU counters are snapshot to the shadow registers. PMUSNAPSHOTREQ must be asynchronous-safe, and operates in a four-phase handshake with PMUSNAPSHOTACK.	Connect to external debug and trace control 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 four-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.	
TSVALUEB[63:0]	Input	Global system timestamp value in binary format	Connect to external system timestamp counter output

B.12 Interrupt and event signals

The following table shows the interrupt and event signals.

All signal names in this section are root names, which are specified as RootName. CMN-700 interfaces use RootName within a fully specified signal name as follows:

CMN-700 interface signal name == RootName_NID#, where # represents the node ID corresponding to the specific interface.



All interrupts are considered active high level sensitive.

Signal definitions

Table B-33: Interrupt and event signals

Signal	Direction	Description	Connection information
INTREQPPU	Output	Power state transition complete	Connect to external interrupt control logic or Generic Interrupt Controller
INTREQPMU_NID<x>	Output	PMU count overflow interrupt. NID indicates node ID where <x> represents the NodeID number for the HN-D/DTC or HN-T/DTC.	
INTREQERRNS	Output	Non-secure error handling interrupt	
INTREQERRS	Output	Secure error handling interrupt	

Signal	Direction	Description	Connection information
INTREQFAULTNS	Output	Non-secure fault handling interrupt	
INTREQFAULTS	Output	Secure fault handling interrupt	
INTREQMPAMERRNS	Output	Non-secure *Memory Partitioning And Monitoring* (MPAM) fault handling interrupt	
INTREQMPAMERRS	Output	Secure MPAM fault handling interrupt	

B.13 MBIST interface signals

Signals that support MBIST capabilities are included in CMN-700.

Signal definitions

Table B-34: MBIST signals

Signal	Direction	Description	Connection information
nMBISTRESET	Input	Primary reset to enter MBIST. Active-LOW. Must be HIGH during functional non-MBIST operation.	Tie HIGH if unused
MBISTREQ	Input	SLC MBIST mode request	Tie LOW if unused

B.14 Power management signals

The following contains information on power management signals for the logic power domain.

If PACTIVE_LOGIC is asserted, the system cannot be powered down.

Signal definitions

Table B-35: Power management signals for logic power domain

Signal	Direction	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[4:0]	Input	The power state to which a transition is requested.	Connect to external power management controller or tie to `5'b01000` if unused
PACCEPT_LOGIC	Output	Indicates acknowledgment of the power state transition and completion of the power state transition within CMN-700	Connect to external power management controller

[8] If *Multicycle Path* (MCP), the MCP duration must be ≤8 cycles to the last flop to receive this signal. This constraint is a requirement for implementation.

Signal	Direction	Description	Connection information
PDENY_LOGIC	Output	Indicates denial of the power state transition.	
PACTIVE_LOGIC	Output	Hint that indicates activity across the CMN-700. When LOW, indicates the possibility of entering static retention or the OFF state.	

B.15 Processor event interface signals

Signals that support processor event interface capabilities are included in CMN-700. Processor event interface signals are present at RN-F, RN-I, and RN-D node locations.

All signal names in this section are only a root name indicated as RootName. CMN-700 interfaces use RootName within a more fully specified signal name as follows:

CMN-700 interface signal name == RootName_NID#, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-36: Processor event interface signals

Signal	Direction	Description	Connection information
EVENTIREQ	Output	Event input request for processor wake up from WFE state. Remains asserted until EVENTIACK is asserted, and is not reasserted until EVENTIACK is LOW.	Connect to EVENTIREQ input of processor
EVENTIACK	Input	Event input request acknowledge. Must not be asserted until EVENTIREQ is HIGH, and then must remain asserted until after EVENTIREQ goes LOW.	Connect to EVENTIACK output of processor, or tie to EVENTIREQ output of CMN-700 if unused
EVENTOREQ	Input	Event output request for processor wake up, triggered by SEV instruction. Must only be asserted when EVENTOACK is LOW, and then must remain HIGH until after EVENTOACK goes HIGH.	Connect to EVENTOREQ output of processor, or tie LOW if unused
EVENTOACK	Output	Event output request acknowledge. Is not asserted until EVENTOREQ is HIGH, and then remains asserted until after EVENTOREQ goes LOW.	Connect to EVENTOACK input of processor

1. Event handling logic external to CMN-700 must handle EVENT_OUT from CHI processor. EVENT_OUT is a multicycle pulse. If system integration wants to connect the EVENT_OUT to CMN-700 EVENTOREQ or EVENTOACK, then stitching logic is required. It is the responsibility of the integrator to design the necessary logic to stitch EVENT_OUT to the four-phase handshake pair, accounting for the asynchronous domain crossing.
2. Event handling logic external to CMN-700 can drive EVENT_IN of CHI processor.

Appendix C Revisions

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

C.1 Revisions

This section lists the differences between released versions of the document.

Table C-1: Issue 0000-01

Change	Location
First release	-

Table C-2: Differences between issue 0000-01 and issue 0001-02

Change	Location
Improvements to descriptions	Throughout the document

Table C-3: Differences between issue 0001-02 and issue 0100-03

Change	Location
Improvements to descriptions	Throughout the document
New section	7.9 CCG performance events on page 1257
New section	B.10 CXS interface signals on page 1296
New publication added	About this book. Other publications
New subsection	2.2 Compliance on page 19
Moved "Top-level configurable options"	2.5.3 Device placement and configuration on page 30
References to "Single MXP" removed	Throughout the document
"CML" replaces "CXG and CCG"	3.1.1.7 CML on page 45
New subsection added	3.1.3.2 CML Port Aggregation Groups on page 65
New subsection added	3.1.1.17 CXS Domain Bridge on page 55
Table added "RN SAM"	3.4.6.9 Address bit masking in the RN SAM on page 138
Table added "HN SAM"	3.4.7.4 HN-F SLC and SF flexibile addressing on page 160
Removed "CHI node IDs for Single MXP configurations"	3.4.2 Node ID mapping for configurations with extra device ports on page 106
Moved sections "RN SAM target ID selection", "DVM target ID", and "GIC target ID"	3.4.6.1 Target IDs on page 112
New information added to HN-F SAM section	3.4.7 HN-F SAM on page 147
Moved section "CXHA error handling" to under "CML error handling"	3.8.10 CCG error handling on page 226
New subsection added	3.8.10 CCG error handling on page 226
New information added to "HN-F CBusy"	3.9.5 Completer Busy indication on page 235
New information added to "Advanced Cbusy handling"	3.9.5.1 Advanced CBusy handling in HN-F on page 237
New section	3.9.10 REQ and DAT RSVDC for CML non-SMP links on page 244

Change	Location
New section	4.1.6 APB-only access on page 256
New section	4.2.1 APB register summary on page 259
New information added to "Identifying clusters and individual devices in clustered mode - HA register"	5.2.15 Identifying clusters and individual devices in clustered mode on page 1174
New information added to "POCQ resource allocation"	5.5.2 POCQ resource allocation on page 1193
New subsection added	5.5.2.1 Request retry based on POCQ resource allocation on page 1196
New subsection added	5.5.2.2 Credit grant for retried request on page 1197
New note added to "CHI feature support for CML"	A.4 CHI feature support for CML on page 1266
New table added to "RSVDC signal description"	B.5.4 RSVDC signal description on page 1293

Table C-4: Differences between issue 0100-03 and issue 0200-04

Change	Location
Improvements to descriptions	Throughout the document
Removed references to CXG, CXRA, CXHA, CXLA, CCIX, MTSX and MTU	Throughout the document
"Preface" has been changed to Chapter 1 "Introduction"	1. Introduction on page 13
"Introduction" chapter has been changed to Chapter 2 "What is CMN-700"	2. What is CMN-700? on page 17
New subsection added to section "Network layer functions"	4.4.3.2 SAM programming examples on page 1123
New subsection added to section "Network layer functions"	3.4.14 CMN-700 expanded RAID on page 190
New subsection added to section "SLC memory system components and configuration"	5.2.5 Memory address decode error handling on page 1162
Updates to Crosspoint section: • Update to description in the number of device ports an MXP configuration can support based on the number of mesh ports. • The maximum mesh configuration size changed to 12 x 12	3.1.1.1 Crosspoint on page 40
Update to Component Aggregation Layer section: • Added SNF-E support for CAL4 • Removed reference of RCCCAL2 from section	3.1.1.13 Component Aggregation Layer on page 48
Update to Credited Slices section: • Instances of MCSX and MCSY have been removed and replaced with MCS	3.1.1.14 Credited Slices on page 50
Update to Asynchronous Mesh Credited Slice: • Updated description on how to configuring multiple clock domains • Removed graphic of AMCS topologies	3.1.1.14.2 Asynchronous Mesh Credited Slice on page 53
Update to CML system configurations • Removed reference to PCIe switch in CML topologies	3.1.3 CML system configurations on page 62
Section 3.1.3.4 renamed "Tunneling PCIe through CML_SMP link" • Details added about when Tunneling of PCIe traffic is advantageous • Details about high ordered and low ordered interleave.	3.1.3.4 Tunneling PCIe traffic through CML SMP link on page 66

Change	Location
New sub-section added	3.1.3.5 Streaming PCIe write traffic through CML SMP link on page 68
Section 3.1.4.1 renamed "Dual DAT, REQ, RSP, and SNP CHI channels"	3.1.4.1 Dual CHI channels on page 68
Section 3.1.4.2 renamed "Dual DAT, REQ, RSP, and SNP CHI channel selection"	3.1.4.2 Dual CHI channel selection on page 69
Section 3.1.4.3 renamed "Dual DAT, REQ, RSP, and SNP CHI channel selection registers"	3.1.4.3 Dual CHI channel selection registers on page 71
Update to DSU and DMC AXI5 Utility Bus • Removed sentence about additional amount of AXU bus ports since there is only a maximum of 1.	3.1.5 DSU and DMC AXI5 Utility Bus on page 77
Section 3.2.2 renamed "CCG clock inputs" • Remove graphics related to CXG from section	3.2.2 CCG clock inputs on page 81
Update to HN-F with CAL support • Content added describing "SCG target ID selection with CAL mode power-of-two hashing" • Content added describing "SCG target ID selection with CAL mode non-power of two and hierarchical hashing"	3.4.6.6 HN-F with CAL support on page 132
Update to section SAM support for CML Port Aggregation • Content added describing "AXID based CML port aggregation"	3.4.6.8 SAM support for CML Port Aggregation on page 135
Update to section HN-F SAM • Update to graphic "HN-F SAM target ID selection policy" • Content added to show support of 2, 4 and 8 SN mode in the default hashed region	3.4.7 HN-F SAM on page 147
Update to subsection section 3.4.7.4: • Content added to show support of setaddr shuttering • Added note for restriction when using setaddr shuttering	3.4.7.4 HN-F SLC and SF flexibile addressing on page 160
Update to section Reliability, Availability, and Seviceability: • Removed top-level error diagram	3.8 Reliability, Availability, and Serviceability on page 208
Update to section HN-F error handling • Added content about Memory Address Decode Error and Error Handling	3.8.5 HN-F error handling on page 216
Section CML error handling replaced with CCG error handling	3.8.10 CCG error handling on page 226
Update to section Support for early completion of DVMOp requests: • Content added with details about DVMOps message broadcast • New table added describing the DVM message broadcasts	3.9.4.1 Support for early completion of DVMOp requests on page 231
Update to section Register Summary • New registers added to section • por_hnf has been replaced with cmn_hns	4.2 Register summary on page 259
Update to sub section CCG performance events • Added note to inform customer about of size of counter width for the equation	7.9 CCG performance events on page 1257
Update to sub section 7.10 • New note added to explain CCG HA's PB allocation events	7.10 Occupancy and lifetime measurement using PMU events on page 1261

Table C-5: Differences between issue 0200-04 and issue 0300-05

Change	Location
Improvements to descriptions	Throughout the document
Updated tables in section 2.5.2 Mesh sizing and top-level configuration	2.5.2 Mesh sizing and top-level configuration on page 25
Updated tables in section 2.5.3 Device placement and configuration	2.5.3 Device placement and configuration on page 30
New content added to section 3.1.1.13 Component Aggregation Layer • CALBYP2 • CALBYP4 • HCAL2 and HCALBYP4	3.1.1.13 Component Aggregation Layer on page 48
New content added to section 3.1.4.1 Dual CHI channels • Added `EN_2X_REQ_VC` and `EN_2X_SNP_VC` parameters	3.1.4.1 Dual CHI channels on page 68
New content added to section 3.1.4.2 Dual CHI channel selection • Paragraph added explaining the REQ channel	3.1.4.2 Dual CHI channel selection on page 69
New content added to section 3.1.4.3 Dual CHI channel selection registers • Added subsection Replicated channels	3.1.4.3 Dual CHI channel selection registers on page 71
New content added to section 3.4.5 RN SAM	3.4.6 RN SAM on page 112
New content added to section 3.4.5.8 SAM support for CML Port Aggregation	3.4.6.8 SAM support for CML Port Aggregation on page 135
New Content added to section 3.4.15 CMN-700 expanded RAID	3.4.14 CMN-700 expanded RAID on page 190
Added new sections section 3.9.4.2 Optimization of DVMOps and DVMSyncs	3.9.4.2 Optimization of DVMOps and DVMSyncs on page 232
Added new sections section 3.9.4.3 Remote DVM Optimization based on VMID filtering	3.9.4.3 Remote DVM Optimization based on VMID filtering on page 234
Added new sections section 3.9.4.3.1 VMID filtering Optimization	3.9.4.3.1 VMID Filtering Optimization on page 234
New Content added to section 3.9.1.4 Atomic requests in RN-I and RN-D	3.9.1.4 Atomic requests in RN-I and RN-D on page 228
Update to section 4.2 Register Summary • New registers added to section • por_hnf has been replaced with cmn_hns	4.2 Register summary on page 259
New section added 6.4 PMU system programming	4.4.9 PMU system programming on page 1154
New section added 7.1.2 RN-I and RN-D write data cancel	7.1.2 RN-I and RN-D write data cancel on page 1228

Table C-6: Differences between issue 0300-05 and issue 0301-06

Change	Location
Improvements to description	Throughout the documentation

Table C-7: Differences between issue 0301-06 and issue 0302-07

Change	Location
Improvements to description	Throughout the document
Updates to content for inclusive language	Throughout the document
New table added in section 7.1.5 for HN-P	7.1.5 CMN buffer lifetime and recommended parameter settings on page 1230
New details added for the behavior of s*dis_data_interleaving and the description of the new behavior	7.1 Performance optimization guidelines on page 1225

Change	Location
New details added s*dis_data_interleaving to RN-X aux_ctl registers	4.3.14.7 por_rnd_aux_ctl on page 974 4.3.15.7 por_rni_aux_ctl on page 997
SAM programming examples removed from Functional Description to the CMN-700 programming in the Programmers Model	4.4.3.2 SAM programming examples on page 1123