Arm® Neoverse™ CMN-650 Coherent Mesh Network

Revision: r2p0

Technical Reference Manual

Non-Confidential

Issue 04

101481_0200_04_en

Arm® Neoverse™ CMN-650 Coherent Mesh Network

Technical Reference Manual

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-00	12 June 2019	Confidential	First Limited Access release for r0p0
0100-01	6 November 2019	Confidential	First Early Access release for r1p0
0200-02	17 June 2020	Confidential	First Early Access release for r2p0
0200-03	7 May 2021	Confidential	Second Early Access release for r2p0
0200-04	11 August 2021	Non-Confidential	First Full release for r2p0

Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND

REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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

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

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

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

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.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

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, `r`*`1`*`p`*`2`*, where:

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

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

1.2 Intended audience

This 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-650 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.

Typographic conventions

Convention	Use
italic	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace bold`	Denotes language keywords when used outside example code.
`monospace underline`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

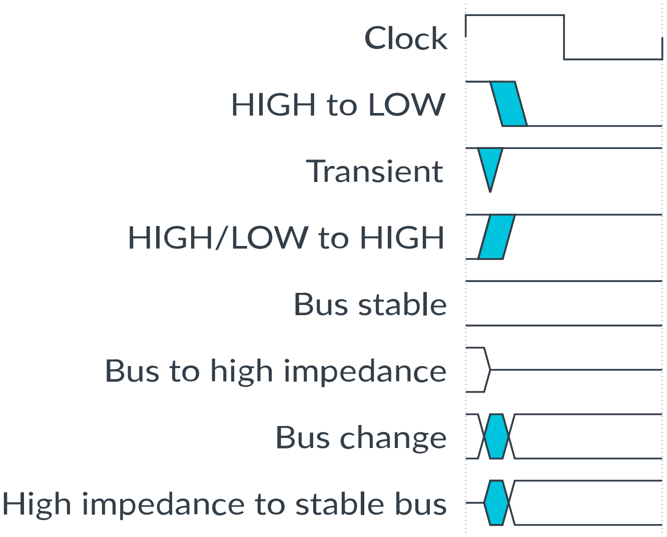
Convention	Use
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
⚠ Caution	This represents a recommendation which, if not followed, might lead to system failure or damage.
⚠ Warning	This represents a requirement for the system that, if not followed, might result in system failure or damage.
⚠ Danger	This represents a requirement for the system that, if not followed, will result in system failure or damage.
✎ Note	This represents an important piece of information that needs your attention.
💡 Tip	This represents a useful tip that might make it easier, better or faster to perform a task.
📌 Remember	This is a reminder of something important that relates to the information you are reading.

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 Additional reading

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

Table 1-2: Arm Publications

Document name	Document ID	Licensee only
AMBA® AXI and ACE Protocol Specification	IHI 0022	No
AMBA® CXS Protocol Specification	IHI 0079	No
AMBA® Low Power Interface Specification Arm® Q-Channel and P-Channel Interfaces	IHI 0068	No
AMBA® 4 AXI4-Stream Protocol Specification	IHI 0051A	No
AMBA® 5 CHI Architecture Specification	IHI 0050	No
Arm® CoreSight™ Architecture Specification v3.0	IHI 0029E	No

Document name	Document ID	Licensee only
Arm® Architecture Reference Manual Armv7-A and Armv7-R edition	DDI 0406	No
Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile	DDI 0487	No
Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A	DDI 0598	No
Arm® Reliability, Availability, and Serviceability (RAS) Specification Armv8, for the Armv8-A architecture profile	DDI 0587	No
Principles of Arm® Memory Maps White Paper	DEN 0001	No
Arm® Neoverse™ CMN-650 Coherent Mesh Network Release Note	PJDOC-1779577084-13193	Yes
Arm® Neoverse™ CMN-650 Coherent Mesh Network Configuration and Integration Manual	101480	Yes
Arm® Socrates™ for Neoverse™ CMN-650 User Guide	101565	Yes
Arm® Neoverse™ N1 hyperscale reference design GIC-600 Integration using CMN-600 AXI4-Stream Interfaces White Paper	PJDOC-1779577084-5931	Yes
Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview	100459	Yes
Arm® CoreLink™ DMC-620 Dynamic Memory Controller Technical Reference Manual	100568	Yes

Table 1-3: Other Publications

Document ID	Document name
JEP106	*Standard Manufacturers Identification Code*, http://www.jedec.org.
-	*Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*, https://www.ccixconsortium.com/

1.5 Feedback

Arm welcomes feedback on this product and its documentation.

Feedback on this product

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

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

Feedback on content

Information about how to give feedback on the content.

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

- The title Arm® Neoverse™ CMN-650 Coherent Mesh Network Technical Reference Manual.
- The number 101481_0200_04_en.
- If applicable, the page number(s) to which your comments refer.

- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

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

2 What is CMN-650?

This chapter introduces CMN-650 which is an AMBA® 5 CHI interconnect with a customizable mesh topology.

2.1 About CMN-650

The CMN-650 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-650 is a scalable mesh interconnect that supports 1-256 processor compute clusters.

You can configure CMN-650 using the Arm® Socrates™ system IP Tooling platform. Socrates™ is an environment for the configuration of Arm IP. Using Socrates™, you can configure the following CMN-650 characteristics:

- Custom interconnect size and device placement
- Optional *System Level Cache* (SLC). For more information about the features of the SLC memory system, see 6.1 About the SLC memory system on page 867.

CMN-650 supports Arm® AMBA® 5 CHI Issue D, including the following features:

- *Memory System Performance Resource Partitioning and Monitoring* (MPAM)
- *Persistent Cache Maintenance Operation* (PCMO) with two part response
- Increased TxnID width
- *Ordered Write Observation* (OWO) flow enhancements
- *Completer Busy* (CBusy) indication

CMN-650 provides system-level alignment by providing the following system functionality:

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

CMN-650 is compatible with the following types of IP:

- *Dynamic Memory Controller* (DMC)
- *Generic Interrupt Controller* (GIC)
- *Memory Management Unit* (MMU)
- Interconnect
- Armv8.0, Armv8.2, and Armv8.4 processors

CMN-650 provides an optional *Coherent Multichip Link* (CML) feature. CML is compliant with the CCIX standard and allows you to support up to four SoCs in a coherent system.

2.2 Compliance

The CMN-650 product is based on Issue D 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-650 supports the *AMBA® 5 CHI Architecture Specification* Issue D, and is also backwards compatible with Issue C and Issue B. For more information about compatibility, see 4.14 Backward compatible RN-F support on page 180.

The CMN-650 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.

CCIX architecture

The CML CCIX implementation is compliant with *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0* dated September 6, 2019.

2.3 Features

The CMN-650 product provides the following key features:

- Highly scalable mesh network topology configurable up to a 10 × 10 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 masters
- Optional *Component Aggregation Layer* (CAL) for device interface port expansion
- Up to 40 SN interfaces
- Up to 36 RN-Is with up to three ACE5-Lite ports each (96 total):

Note

More devices are supported by adding more levels of interconnect hierarchy, such as the CoreLink™ NIC-450 Network Interconnect, to the system.

- Option for a second pair of 256-bit DAT and RSP channels, one for each direction. CMN-650 supports either single 256-bit or dual (2x 256-bit) DAT and RSP channel configurations for each direction, to increase device upload and MXP-MXP bandwidth.
- Optional support for non-XY routing algorithm between specified source-target pairs
- Maximum *Physical Address* (PA) width of 52 bits
- DVM message transport between masters
- 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 64 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 512MB of tag RAM for increased coherency scalability consisting of up to 64 partitions (one per HN-F).
- Up to 16 HN-Is, each with an ACE-Lite master port
- An HN-I that is known as HN-P, which includes PCIe optimizations
- Option for a single synchronous clock domain or four rectangular asynchronous clock domains
- *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) allows for the creation of CMN-650 systems without physical DDR memory.

- RAS features including transport parity, optional data path parity, *Single-Error Correction and Double-Error Detection* (SECDED) ECC, and data poisoning signaling
- Up to ten CCIX ports, supporting one, two, or three CCIX links for chip-to-chip coherent communication. Each CCIX port has a CXS interface that transports CCIX TLPs.
  - CXS interface width options of 256-bit and 512-bit
- Support for CCIX port-to-port forwarding
- *Address Based Flush* (ABF)
- Way-based SLC partitioning
- Source-based way locking
- *AXI4-Stream* (A4S) support (for GIC traffic only)
- *CCIX Slave Agent* (CXSA) support in CML mode
- Support for AXI loopback signaling

2.4 Global configuration parameters

CMN-650 has various global configuration parameters that define the properties and behavior of interconnect. These parameters also set and define the behavior of some of the optional features.

The following table shows the global configuration parameters for CMN-650.

Table 2-1: Global configuration parameters

Parameter	Description	Values (default)	Comments
META_DATA_EN	Enables metadata preservation mode	0, 1 (false)	Enables DAT RSVDC propagation. For more information, see 4.18.7 DAT RSVDC propagation on page 212.
CHI_MPAM_ENABLE	Enables MPAM feature	0, 1 (true)	-
REQ_RSVDC_WIDTH	Width of RSVDC field in REQ flit	4, 8 (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-650 system with RNF_CHIB_ESAM or RNF_CHIC_ESAM devices. For these configurations, CMN-650 supports a maximum *RE\Q_ADDR_WIDTH* value of 48.
PA_WIDTH	System PA width	34, 44, 48, 52 (48)	For more information about permitted combinations of *RE\Q_ADDR_WIDTH* and *PA_WIDTH*, see 2.4.1 Addressing capabilities on page 24.
DATACHECK_EN	Enables Data Check	0, 1 (false)	Data Check refers to data byte parity checking.
NUM_REMOTE_RNF	Number of RN-Fs for CML configurations on the all the remote chips combined	0-384 (0)	-

Parameter	Description	Values (default)	Comments
FLIT_PAR_EN	Enables flit parity	0, 1 (true)	-
RNSAM_NUM_NONHASH_REGION	Number of non-hashed regions supported by the RN SAM	8, 12, 16, 20 (8)	-
RNSAM_NUM_ADD_HASHED_TGT	Number of extra hashed target IDs supported by the RN SAM, beyond the local HN-F count	0, 2, 4, 8, 16, 32 (0)	For configurations where HN-F CALs are present, see 4.5.7 Support for HN-Fs with CAL in the RN SAM on page 110 for further considerations.
RNSAM_NUM_QOS_REGIONS	Number of memory regions for QoS override	0, 4, 8 (0)	-
2XDATRSP_EN	Enables dual DAT and RSP channel feature	0, 1 (false)	-
PORTFWD_EN	Enables CCIX port-to-port forwarding feature	0, 1 (false)	-
XY_OVERRIDE_CNT	Number of source-target MXP pairs whose XY route path can be overridden	0, 2, 4, 8, 16 (0)	-

Related information

- 3.5 Configure CMN-650 on page 50

2.4.1 Addressing capabilities

CMN-650 supports a 34-bit, 44-bit, 48-bit, or 52-bit PA width. The PA width defines the PA space for which read and write transactions are supported in the interconnect. It is configured using Socrates and results in the *PA_WIDTH* global parameter in the CMN-650 RTL.

CHI interfaces in CMN-650 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-650 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 2-2: 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

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

The minimum PA width that CCIX supports is 48 bits. Therefore, a mismatch between address widths can occur in the following type of configuration:

- The CMN-650 *PA_WIDTH* parameter value is less than 48.
- The system is a CML system with non-CMN-650 CCIX components.

This mismatch means that the upper address bits of a CCIX request or snoop from a remote CCIX component might be truncated in CMN-650. For example, in this kind of system, bits [47:34] or [47:44] might be truncated in CMN-650. To prevent this truncation, software must ensure that non-CMN-650 CCIX components do not present requests and snoops with addresses higher than the CMN-650 *PA_WIDTH* to CMN-650.

2.5 Device-level configuration parameters

Each internal device, external CHI device port, and CXLA have a set of configuration parameters. These parameters define the properties and behavior of the device or port and also set up optional features.

The following tables show the configuration parameters for individual CMN-650 devices.

Note

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

Table 2-3: RN-F and SN-F port configurable options

Parameter	Description	Values (default)	Comments
POISON	Enables data poison. RN-F port only.	0, 1 (true)	-
DATACHECK	Enables Data Check. RN-F port only. **Note:** When the global *DATACHECK* parameter is 0, *DATACHECK* must be set to 0.	0, 1 (false)	Enables end-to-end data byte parity
RXBUF_NUM_ENTRIES	Number of receive flit buffers inside CMN-650 on this port. To achieve full bandwidth operation, this number must equal the CHI credit return latency (in cycles) for flit transfers from RN-F or SN-F to the interconnect. **Note:** The credit return latency is one cycle in the interconnect. This value must be added to the credit latency in the RN-F or SN-F to arrive at the total credit return latency.	2-4 (3)	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. For more information, see 4.13 Link layer on page 178.

Table 2-4: RN-I and RN-D configurable options

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 (32)	-
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 (32)	If *NUM_RD_BUF* is 128 or 256, *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, 8, 16, 24, 32, 64, 96, 128, 256 (24)	This value must be 256 when *NUM_RD_REQ* is 256 and less than or equal to *NUM_RD_REQ* for all other cases. *NUM_RD_BUF* > 64 instantiates RAM for the read data buffers.
NUM_PREALLOC_RD_BUF	Number of pre-allocated read data buffers	4, 8, 16, 32 (8)	This value must be ≤ *NUM_RD_BUF*.
AXDATAPOISON_EN	Enables data poison on AXI or ACE-Lite interface	0, 1 (0)	-
AXLOOPBACK_EN	2-bit loopback enable on AXI or ACE-Lite interface	0, 1 (0)	-
AXMPAM_EN	Enables MPAM feature on AXI or ACE-Lite interface	0, 1 (true)	-
FORCE_RDB_PREALLOC	Force read data buffer pre-allocation	0, 1 (0)	-
ID_WIDTH	**AxID** width for slave ports	11, 16, 24, 32 (11)	-

Table 2-5: HN-F configurable options

Parameter	Description	Values (default)	Comments
SLC_SIZE	Size of *System Level Cache* (SLC) slice	0KB, 128KB, 256KB, 512KB, 1MB, 2MB, 3MB, 4MB (2MB)	-
SF_SIZE	Size of *Snoop Filter* (SF) tag RAM slice	512KB, 1MB, 2MB, 4MB, 8MB (4MB)	The optimal total SF size across all HN-Fs is twice the total exclusive cache size for all RN-Fs. For example, for a 32MB total cache size, the recommended SF size is 64MB.
SLC_TAG_RAM_LATENCY	Latency of SLC tag RAM	1, 2, 3 (2)	Valid Tag:Data latency combinations: • 1:2 • 2:2 • 3:3
SLC_DATA_RAM_LATENCY	Latency of SLC data RAM	2, 3 (2)	
NUM_ENTRIES_POCQ	Number of entries in the POCQ tracker	16, 32, 64 (32)	-
SF_RN_ADD_VECTOR_WIDTH	Number of extra bits in the SF to track the RN-Fs	0-127 (0)	-

Parameter	Description	Values (default)	Comments
SF_MAX_RNF_PER_CLUSTER	Maximum number of RN-Fs per cluster as represented in the SF 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)	-
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 monitoring counters	1, 2, 4, 8, 16 (4)	-

Table 2-6: HN-I, HN-D, and HN-T configurable options

Parameter	Description	Values (default)	Comments
AXDATA_WIDTH	Data width on AXI or ACE-Lite interface	128, 256, 512 (128)	-
NUM_AXI_REQS	Number of request tracker entries	8, 32, 64 (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 interface	0, 1 (true)	-

Table 2-7: HN-P configurable options

Parameter	Description	Values (default)	Comments
AXDATA_WIDTH	Data width on AXI or ACE-Lite interface	128, 256, 512 (128)	-
NUM_AXI_REQS	Number of request tracker entries	8, 32, 64 (32)	-
AXDATAPOISON_EN	Enables data poison 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_AXI_REQS	Depth of peer-to-peer write slice AXI request tracker	32, 64 (32)	-
RD_NUM_AXI_REQS	Depth of peer-to-peer read slice AXI request tracker	64, 128, 256 (64)	-

Table 2-8: SBSX configurable options

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	Enables data poison on AXI or ACE-Lite interface	0, 1 (true)	-
AXMPAM_EN	Enables MPAM feature on AXI or ACE-Lite interfaces	0, 1 (true)	-
SBSX_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 2-9: CXG_RA configurable options

Parameter	Description	Values (default)	Comments
RA_NUM_REQS	Depth of request tracker	64, 128, 256 (256)	The number of outstanding requests that an RA can have. Base this number on the round-trip latency of a request completing on a remote chip.
RA_NUM_RDBUF	Depth of read data buffer	16, 24, 32 (16)	The number of CHI read data flits that can be stored in the RA CHI link layer buffer to handle any mesh uploads stalls. For a smaller configuration, the value of 24 is recommended. For a larger configuration, the value of 32 is recommended.
RA_NUM_WRBUF	Depth of write data buffer	16, 24, 32 (24)	The number is based on the round-trip latency of the DBID to data response on CHI. This number can be based on the size of the mesh.
RA_NUM_SNPREQS	Depth of snoop tracker	64, 128, 256 (128)	The number of outstanding snoop credits that are assigned to remote HAs.
RA_NUM_SNPBUF	Depth of snoop data buffer	16, 24, 32 (32)	The number of outstanding CHI snoops to local RN-Fs, including a snoop data buffer per entry.

Table 2-10: CXG_HA device configurable options

Parameter	Description	Values (default)	Comments
HA_NUM_REQS	Depth of request tracker	128, 192, 256 (192)	If passive buffer is not present, then this parameter specifies the following: • The number of CCIX request credits that are assigned or given to remote RAs. • The number of outstanding CHI requests to local CHI HNs. If passive buffer is enabled, then the number of CCIX request credits that are given to remote RAs = *HA_NUM_WRBUF* + *HA_PASS_BUF\F_DEPTH*
HA_NUM_WRBUF	Depth of write data buffer	96, 128 (96)	If passive buffer is not present, then this parameter specifies the number of CCIX data credits that are assigned or given to remote RAs. If passive buffer is present, then the number of CCIX data credits that are given to remote RAs = *HA_NUM_WRBUF* + *HA_PASS_BUFF_DEPTH*.
HA_NUM_SNPREQS	Depth of snoop tracker	96, 128, 256 (96)	The depth of the snoop tracker. The value indicates the number of outstanding snoop requests that an HA can have on CCIX.
HA_NUM_SNPBUF	Depth of snoop data buffer	16, 24, 32 (24)	The number of CHI snoop data flits that can be stored at the HA CHI link layer buffer to handle any mesh uploads stalls. For a smaller configuration, the value of 24 is recommended. For a larger configuration, the value of 32 is recommended.
HA_SSB_DEPTH	Maximum number of remote snoop requests from the local chip that can be sunk at this HA	96, 128, 256 (96)	The value must be at least as large as *HA_NUM_SNPREQS*. The value represents the total number of outstanding snoops that all the CHI HNs can have targeting this particular HA.
HA_PASS_BUFF_DEPTH	Depth of passive buffer	0, 256, 512 (512)	-

Table 2-11: CXLA configurable options

Parameter	Description	Values (default)	Comments
DB_FIFO_DEPTH	FIFO depth in CXLA Domain Bridges (CXDB, PDB)	6, 8 (8)	-
DB_PRESENT	CXLA Domain Bridges (CXDB, PDB) present between CGL and LA	0, 1 (true)	-
CXS_DATA_WIDTH	Width of CXS TX/RX data	256, 512 (256)	When set to 256: • *CXS_MAX_PKT_PER_FLIT_TX* = 2 • *CXS_MAX_PKT_PER_FLIT_RX* = 2 When set to 512: • *CXS_MAX_PKT_PER_FLIT_TX* = 4 • *CXS_MAX_PKT_PER_FLIT_RX* = 4
PORTFWD_EN	Enables CCIX port-to-port forwarding feature	0, 1 (false)	-
PORTFWD_NUM_DYNAMIC_TXBUF	Depth of TX buffer for forwarded CCIX TLPs, which are forwarded using dynamic credits	8, 16, 32 (8)	-
PORTFWD_NUM_STATIC_TXBUF	Depth of TX buffer for forwarded CCIX TLPs, which are forwarded using static credits	8, 16, 32 (8)	-

Related information

- 3.5 Configure CMN-650 on page 50

2.6 Test features

The CMN-650 product includes several test features.

See the *Arm® Neoverse™ CMN-650 Coherent Mesh Network Configuration and Integration Manual* for information about the test features.

2.7 Product documentation and design flow

The CMN-650 product manuals support the design flow process.

Documentation

The following documentation supports the CMN-650 product:

Technical Reference Manual

The *Technical Reference Manual* (TRM) describes the functionality, and how functional options affect the behavior of CMN-650. 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-650 product, contact the following people:

- The implementer to determine:
    - The build configuration of the implementation
    - What integration, if any, was performed before implementing the CMN-650 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-650 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 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 *User Guide* (UG) describes how to use Socrates™ to configure and integrate a custom mesh interconnect.

Note

The UG is part of the Socrates™ product download bundle.

Design flow

CMN-650 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-650 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-650 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-650 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-650 product by programming particular values into registers. The register configuration affects the behavior of the CMN-650 product.

Note

This manual 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 one that software has also configured.

2.8 Product revisions

This section describes the differences in functionality between successive product revisions of the CMN-650 product.

r0p0

First release.

r1p0

Added the following functionality:

- AMBA® Domain Bridge
- APB interface
- 512-bit CXS
- 512-bit HN-I
- Multiple asynchronous clock domains through AMCS
- HN-I CAL
- SBSX CAL

- Non-XY routing override
- Dual DAT/RSP
- RN SAM QoS override
- HN-F CBusy enhancements
- Remote HN-I exclusive support

r1p1

Added the following functionality:

- PrefetchTgt support for range-based address regions in RN SAM

r1p2

Erratum fix release

r2p0

Added the following functionality:

- Mesh size up to 10x10
- CAL2 for HN-P, RN-I, and RN-D
- CAL4 for CHI-D RN-F ESAM
- HN-P
- PCIe local bandwidth with no read burst preservation
- Early DVM completion, including for CXG
- No interleaving of RDATA requirement for RN-I
- Optional dedicated RN-I or RN-D resources per AXI port
- Software configurable MPAM override
- Support for 5-SN hashing in HN-F SAM and RN SAM
- CCIX port to port forwarding
- Support for up to 512 CXRAs with no RAID aliasing and 256 RN-Fs on a single chip
- Increased number of CCIX protocol credits and flit buffer RAMs
- Support for CleanSharedPersistSep and two-part response on SMP CCIX link
- 24-bit metadata preservation in SLC and across CML

3 Components and configuration

This chapter describes the structure of a CMN-650 interconnect. It also describes the internal and external components and things to consider when you configure the interconnect.

You must use the Socrates™ IP Tooling platform to configure a CMN-650 interconnect instance. This TRM does not provide information about how to use Socrates™. For information about using Socrates™, see the *Arm® Socrates™ for Neoverse™ CMN-650 User Guide*, which is bundled with the tool.

3.1 Structure of a CMN-650 interconnect

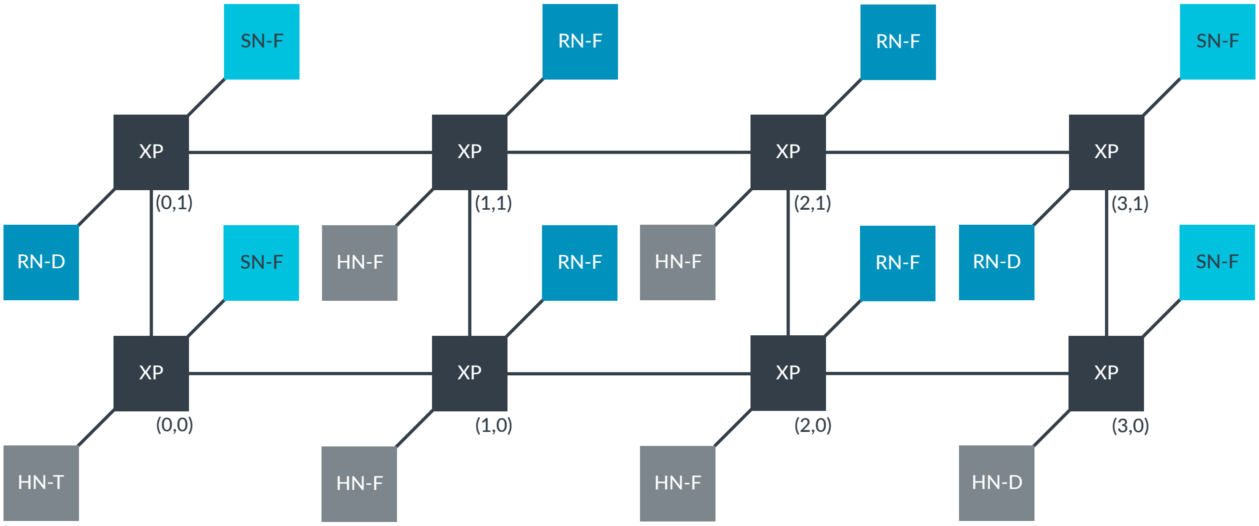
The structure of a CMN-650 interconnect is always an arrangement of *Crosspoints* (XPs) and CHI-compliant devices. XPs connect together to form a mesh, and CHI-compliant devices connect to device ports on the XPs. When configuring CMN-650, you create a topology from these building blocks, according to your system requirements.

A CMN-650 interconnect is a network that is made up of two types of component: CHI devices and XPs. CHI devices generate and receive network requests and responses, and process data. CHI devices connect to XPs through XP device ports. XPs are network routers, which send requests, responses, and data in packets between devices on an XP, or to another XP in the network.

XPs connect horizontally and vertically to each other to form a two-dimensional mesh structure. Each XP has two device ports for connecting external CHI-compliant devices, for example a CHI processor cluster, or internal CMN-650 devices.

For example, the following figure shows a 4 × 2 mesh configuration containing various types of external and internal CHI devices.

Figure 3-1: Example 4 × 2 mesh configuration



Internal CMN-650 devices have various functions. For example, some of these devices have AXI and ACE-Lite interfaces to attach external AXI and ACE-Lite hardware. These devices form bridges between the external hardware and CMN-650, which is a CHI interconnect.

Related information

- 2.2 Compliance on page 21
- 3.3 External interfaces on page 37
- 3.4 Components on page 39
- 3.5 Configure CMN-650 on page 50

3.2 Crosspoint (XP)

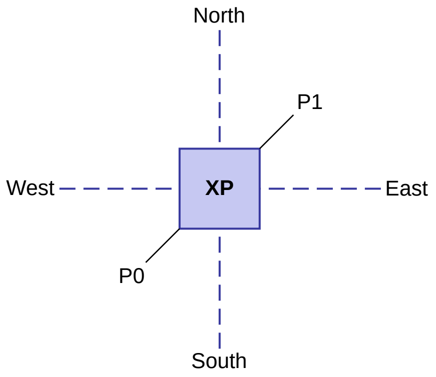
The *crosspoint* (XP) is a switch or router logic module. It is the fundamental building block of the CMN-650 transport mechanism.

Note

The terms XP, *Mesh Crosspoint* (MXP), and *Super Mesh Crosspoint* (SMXP) are used interchangeably throughout this TRM.

The CMN-650 mesh interconnect is built using a set of XP modules. Each XP can have six ports: four mesh ports and two device ports. The following figure shows these ports.

Figure 3-2: Crosspoint (XP)



The dashed lines in the preceding figure represent mesh ports. Each XP can connect to up to four neighboring XPs through mesh ports. Each XP also has two device ports, P0 and P1, for connecting devices.

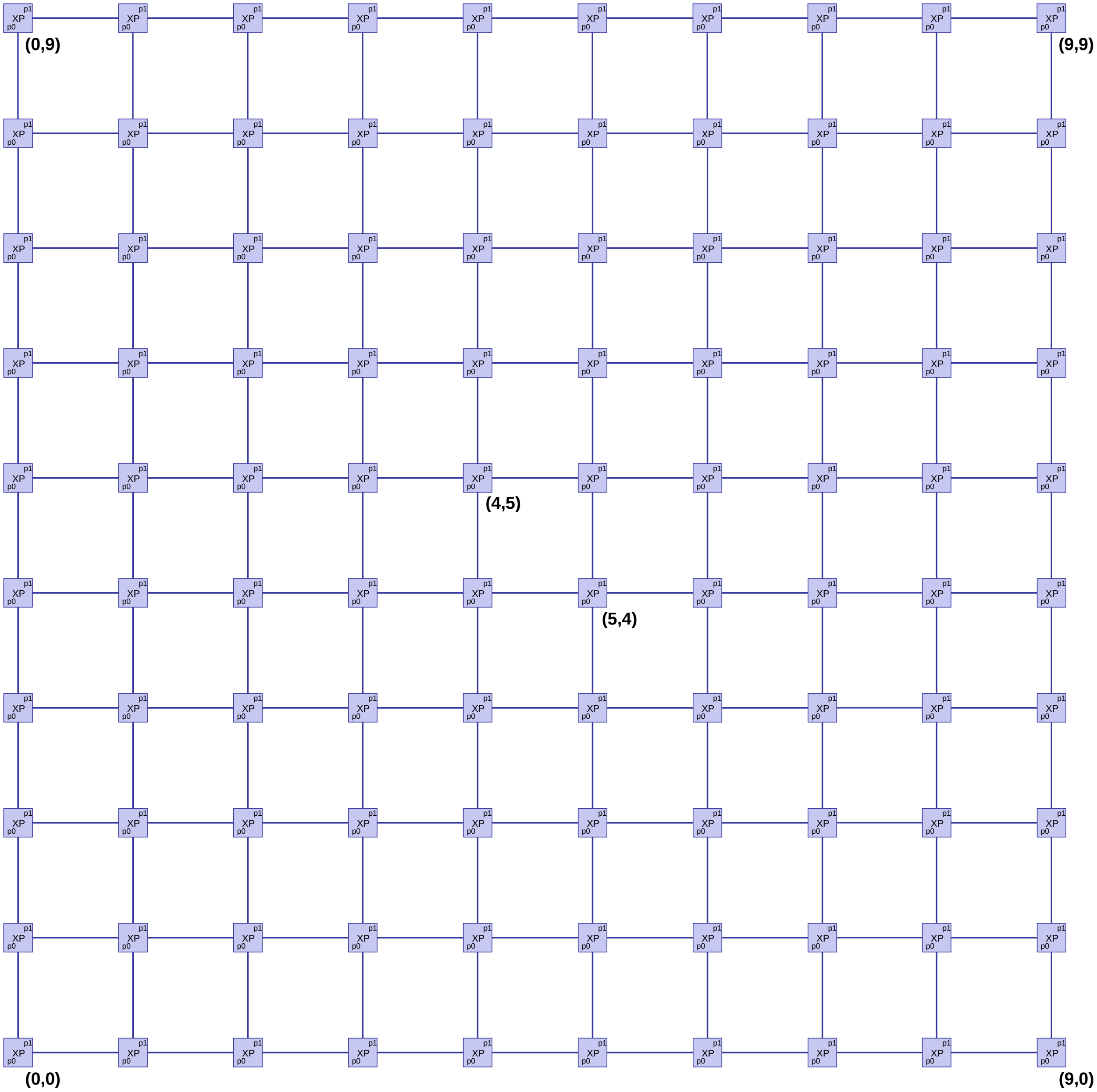
Each XP supports four CHI channels for transporting flits across the mesh from a source device to a destination or target device:

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

The XP modules are arranged in a two-dimensional rectangular mesh topology.

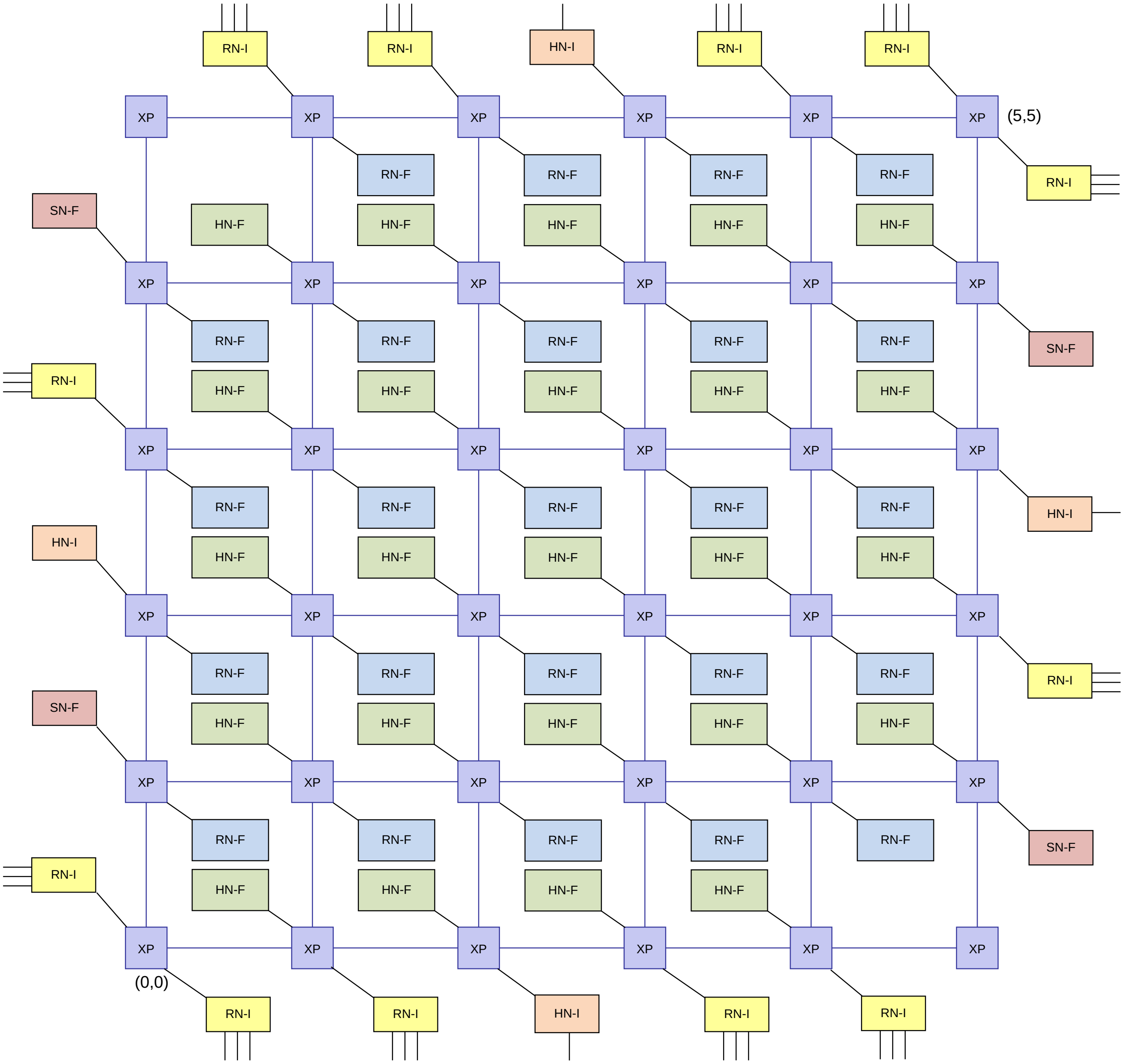
The maximum size for the CMN-650 mesh is 100 XPs arranged in a 10 × 10 grid. Each XP in the grid is referenced using an (X,Y) coordinate system. (0,0) represents the bottom-left corner, and a maximum coordinate of (9,9) represents the upper-right corner. The following figure shows the maximum 10 × 10 mesh configuration with some (X,Y) coordinate values.

Figure 3-3: 10 × 10 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



The x and y coordinates of an XP are also known as the XID and YID respectively.

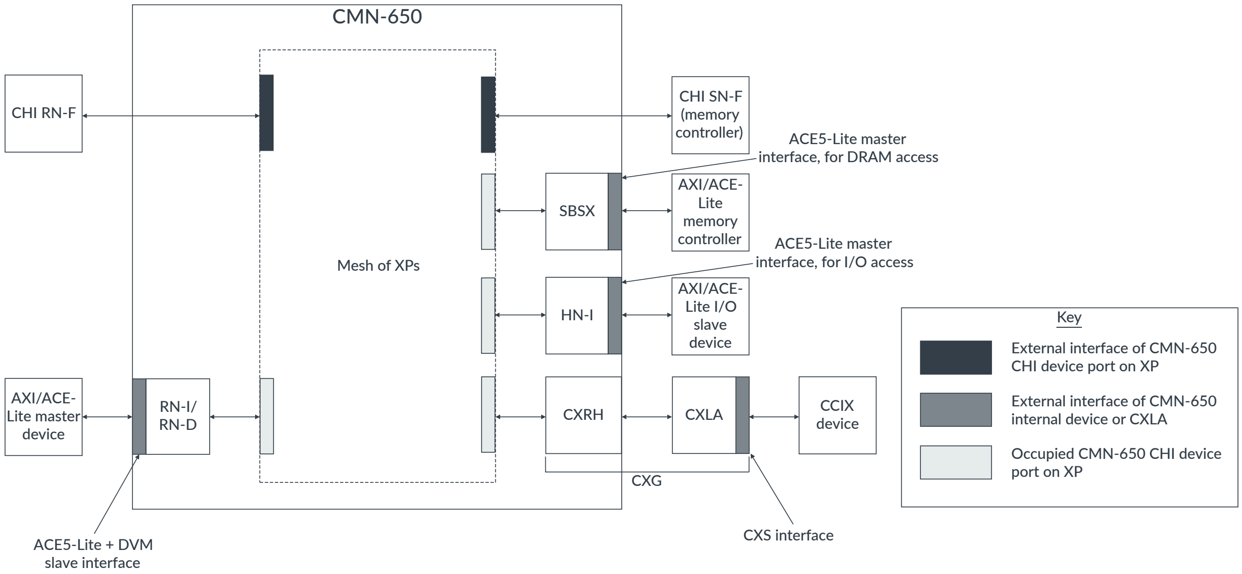
3.3 External interfaces

You can connect external CHI-compatible hardware, such as a processor cluster or memory controller, directly to CMN-650 through an XP device port. Alternatively, CMN-650 has various internal devices with external interfaces for connecting non-CHI external hardware.

The following figure shows the types of external interfaces that the CMN-650 product has.

> **Note**
>
> The figure shows one instance of each interface type, but the number of each type is configurable according to your system requirements. For more information about the supported number of each interface, see Table 3-1: CHI external interface type use, location, and permitted number on page 38.

Figure 3-5: CMN-650 external interfaces


CMN-650
CHI RN-F
CHI SN-F (memory controller)
ACE5-Lite master interface, for DRAM access
SBSX
AXI/ACE-Lite memory controller
Mesh of XPs
ACE5-Lite master interface, for I/O access
HN-I
AXI/ACE-Lite I/O slave device
Key
External interface of CMN-650 CHI device port on XP
External interface of CMN-650 internal device or CXLA
Occupied CMN-650 CHI device port on XP
AXI/ACE-Lite master device
RN-I/ RN-D
CXRH
CXLA
CCIX device
CXG
ACE5-Lite + DVM slave interface
CXS interface

The following table shows information about the use, location, and permitted number of each type of CMN-650 external interface.

Table 3-1: CHI external interface type use, location, and permitted number

External interface type	Use	Location	Permitted number
Interface on CHI device port	Connect native CHI *Fully coherent Request Nodes* (RN-Fs). RN-Fs are master devices with hardware-coherent caches.	XP	1-256
	Connect native CHI *Fully coherent Slave Nodes* (SN-Fs), such as DRAM memory controllers.		1-40 SN-Fs and SBSXs are counted together, meaning that CMN-650 supports a maximum combined total of 40 of these interfaces.
ACE5-Lite master interface, for DRAM access	Connect AXI or ACE-Lite DRAM memory controllers.	*CHI to AXI or ACE-Lite bridge* (SBSX)	

External interface type	Use	Location	Permitted number
ACE5-Lite master interface, for I/O access	Connect AXI or ACE-Lite slave devices, such as an I/O slave or subsystem containing multiple I/O slaves.	HN-I or HN-I variant	1-16
CXS interface	Connect CCIX compatible devices, for example another CMN-650 mesh.	CXG	0-10
ACE5-Lite + DVM slave interface	Connect AXI or ACE-Lite master devices, that do not contain hardware-coherent caches.	RN-I or RN-D	3-96

Related information

- 3.4 Components on page 39
- 3.6 System component selection on page 51

3.4 Components

CMN-650 is made up of various types of devices with different functionality, including router modules, CHI nodes, and bridges. The full list of components that you need depends on the requirements of your system. Some components are optional or only used if certain requirements are met.

Note

CMN-650 can be integrated into a complete SoC system that includes devices that this section does not describe.

External devices

You can connect CHI *Fully coherent Requesting Nodes* (RN-Fs) and *CHI Slave Nodes* (SN-Fs) as external system components to the CMN-650 interconnect through device ports. The CMN-650 product does not include these components, but you must specify them when you are building your interconnect topology.

The following table shows the external device types that CMN-650 supports.

Table 3-2: Supported external devices

Device	Description
RNF_CHIB_ESAM	A CHI Issue B-compliant RN-F without a built-in SAM. RN-Fs are processors, clusters, GPUs, or other RNs with a coherent cache. ESAM-type RN-Fs do not have a built-in SAM, and the SAM logic is contained within CMN-650. For more information about backward-compatible RN-F support and the SAM, see 4.14 Backward compatible RN-F support on page 180 and 4.4 System Address Map (SAM) on page 95.
RNF_CHIC_ESAM	A CHI Issue C-compliant RN-F without a built-in SAM. For more information about backward-compatible RN-F support and the SAM, see 4.14 Backward compatible RN-F support on page 180 and 4.4 System Address Map (SAM) on page 95.

Device	Description
RNF_CHID_ESAM	A CHI Issue D-compliant RN-F without a built-in SAM. For more information about backward-compatible RN-F support and the SAM, see 4.14 Backward compatible RN-F support on page 180 and 4.4 System Address Map (SAM) on page 95.
CHI Slave Node (SN-F)	SN-Fs are CHI memory controllers. SN-Fs are devices which solely receive CHI commands, limited to fulfilling simple read, write, and CMO requests targeting normal memory. CMN-650 supports SN-Fs with a native CHI-C or CHI-D SN interface.

Internal CMN-650 devices

CMN-650 contains several CHI-compliant device components with different functionality. These components connect to the device ports on XPs. You can use some of these devices to connect certain types of non-CHI master and slave devices to CMN-650.

The following table lists the device types that are supplied with CMN-650.

Table 3-3: CMN-650 device types

Device	Description
IO coherent Request Node (RN-I)	Use an RN-I to connect one or more AXI or ACE-Lite master devices to CMN-650. RN-Is have 3 external AXI or ACE-Lite slave interfaces for connecting AXI or ACE-Lite master devices to, and bridges between AXI/ACE-Lite and CHI protocols. Within the interconnect, the RN-I is a non-caching I/O-coherent master device. Therefore, it acts as a CHI RN-I proxy for each upstream AXI or ACE-Lite master device. There is no capability to issue snoop transactions to RN-Is. For more information about the RN-I, see 3.4.1 I/O coherent Request Node (RN-I) and I/O coherent Request Node with DVM support (RN-D) on page 42.
IO coherent Request Node with DVM support (RN-D)	A subtype of RN-I with an ACE-Lite-with-DVM slave interface, allowing the RN-D to accept *Distributed Virtual Memory* (DVM) messages on the snoop channel
Fully coherent Home Node (HN-F)	The HN-F acts as a *Home Node* (HN) for a coherent region of memory. HN-Fs accept coherent requests from RN-Fs and RN-Is, and generate snoops to all applicable RN-Fs in the system as required to support the coherency protocol. HN-Fs are typically configured with one or both of the internal SLC and SF components. The SLC acts as a last-level cache and the SF tracks cachelines that RN-Fs in the system have cached. HN-Fs also contain the combined *Point-of-Serialization and Point-of-Coherency* (PoS and PoC), which is responsible for ordering of all memory requests sent to the HN-F. For more information about the HN-F and its internal components, see 3.4.2 Fully coherent Home Node (HN-F) on page 42.
IO coherent Home Node (HN-I)	Use HN-Is to connect non-coherent I/O slaves or subsystems to CMN-650. HN-Is have an external AXI or ACE-Lite master interface and act as an HN for a downstream I/O slave or subsystem. They are responsible for ensuring proper ordering of requests targeting the slave. HN-Is do not support caching of any data read from or written to the downstream I/O slave or slave subsystem. For more information about this topic and more general information about the HN-I, see 3.4.3 I/O coherent Home Node (HN-I) on page 43.

Device	Description
IO coherent Home Node with Debug Trace Controller (HN-T)	A subtype of HN-I with a built-in *Debug Trace Controller* (DTC) and ATB. For more information about the DTC, see Debug Trace Controller (DTC) on page 44.
IO coherent Home Node with DVM node (HN-D)	A subtype of HN-I with the following built-in components: • DTC • *DVM Node* (DN) • *Configuration Node* (CFG) • Global Configuration Slave • *Power/Clock Control Block* (PCCB) **Note:** Exactly one HN-D is required per CMN-650 instance. The HN-D has an AXI or ACE-Lite external master interface, an APB interface, and supports ATB. For more information about the DTC, CFG, and DTC, see the following sections: • Debug Trace Controller (DTC) on page 44 • Configuration Node (CFG) on page 44 • Power/Clock Control Block (PCCB) on page 45
IO coherent Home Node with PCIe optimization (HN-P)	A subtype of HN-I with PCIe optimizations and dedicated trackers for peer-to-peer PCIe traffic. Only use HN-P for connection to PCIe slaves. HN-Ps have an external AXI or ACE-Lite master interface.
CHI to AXI or ACE-Lite bridge (SBSX)	Use an SBSX to connect an AXI or ACE-Lite slave memory device to CMN-650. For example, you can use an SBSX to connect the CoreLink™ DMC-400 Dynamic Memory Controller to a CMN-650 system. SBSXs have an external AXI or ACE-Lite master interface for connecting an AXI or ACE-Lite slave to, and bridges between CHI and AXI/ACE-Lite protocols. They convert and forward simple CHI read, write, and CMO commands to the slave memory device. For more information about the SBSX, see 3.4.4 AMBA 5 CHI to ACE5-Lite bridge (SBSX) on page 45.
CCIX Gateway (CXG)	A CXG is a bridge between CHI and a *CCIX Port* (CXS), which enables formation of CML systems. The internal component of the CXG within the CMN-650 hierarchy is the CXRH, which includes RA and HA functionality. **Note:** The CXRH connects to a CXLA, which is external to the CMN-650 hierarchy, to form a CXG. For more information about the CXG, see 3.4.5 CCIX Gateway (CXG) on page 45.

Mesh components

CMN-650 includes various internal components that you can use to customize the structure of the interconnect.

The following table shows the mesh components that CMN-650 supports.

Table 3-4: CMN-650 mesh components

Component	Description
Crosspoint (XP)	A switch or router logic module. XPs are the fundamental building block of the CMN-650 transport mechanism. XPs connect together through mesh ports. Devices connect to the mesh through device ports on XPs. For more information about the XP, see 3.2 Crosspoint (XP) on page 34.

Component	Description
Component Aggregation Layer (CAL)	Allows multiple devices to connect to a single device port on an XP. Only certain devices can connect to specific types of CAL. All devices that connect to a single CAL must be of the same type and you must configure them identically. There are also bypass variants of each CAL type. For more information, see 3.4.6 Component Aggregation Layer (CAL) on page 46.
Credited Slices (CSs)	Credited register slices that incur latency in communication but help with timing closure. There are various types of CSs, which are used for different parts of the interconnect. For more information, see the following sections: • 3.4.7 Credited Slices (CSs) on page 47 • 3.4.7.1 Mesh Credited Slice (MCS) on page 48 • 3.4.7.2 Asynchronous Mesh Credited Slice (AMCS) on page 48 • 3.4.7.3 Device Credited Slice (DCS) on page 50 • 3.4.7.4 CAL Credited Slice (CCS) on page 50
CHI Domain Bridge (CDB)	Bridges two CHI interfaces that operate in two different clock domains, power/voltage domains, or both. For more information about the CDB, see the *Arm® Neoverse™ CMN-650 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.
AMBA Domain Bridge (ADB)	Bridges two AXI, ACE5-Lite, or ACE5-Lite-with-DVM interfaces that operate in two different clock domains, power/voltage domains, or both. For more information about the ADB, see the *Arm® Neoverse™ CMN-650 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.

Related information

- 3.8 Permitted numbers of devices and system resources in the mesh on page 55

3.4.1 I/O coherent Request Node (RN-I) and I/O coherent Request Node with DVM support (RN-D)

The *I/O-coherent Request Node* (RN-I) and *I/O coherent Request Node with DVM support* (RN-D) connect I/O-coherent AMBA masters to the rest of the CMN-650 system.

An RN-I bridge includes three ACE-Lite slave interfaces. An RN-D bridge includes three ACE-Lite-with-DVM slave ports.

The RN-I and RN-D bridges can act as a proxy only for masters that do not contain hardware-coherent caches. There is no capability to issue snoop transactions to RN-Is or RN-Ds.

3.4.2 Fully coherent Home Node (HN-F)

HN-Fs are responsible for managing the coherent part of the system address space. HN-Fs have various subcomponents, which support its function as the HN for coherent memory. Some of these components are required and some are optional.

Each HN-F in the system is configured to manage a specific portion of the overall address space. The entire DRAM space is managed through the combination of all HN-Fs in the system.

The HN-F consists of the following components:

SLC

The 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.

Combined PoS and PoC

The combined PoS and PoC are 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 cache line, and request ordering as required by the RN-F.

SF

The SF tracks cache lines that are present in the RN-Fs. Using the SF 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.

Note

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

3.4.3 I/O coherent Home Node (HN-I)

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

The HN-I acts as a proxy for all the RNs of CMN-650, 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 slave 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.

Caution

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

CMN-650 also has the following HN-I variants with extra functionality:

IO coherent Home Node with Debug Trace Controller **(HN-T)**

HN-I subtype with a built-in *Debug Trace Controller* (DTC) and ATB.

IO coherent Home Node with DVM node **(HN-D)**

HN-I subtype with the following built-in components:

- DTC
- *DVM Node* (DN)

  - *Configuration Node* (CFG)
  - Global configuration slave
  - *Power/Clock Control Block* (PCCB)

***IO coherent Home Node with PCIe optimization* (HN-P)**

HN-I subtype with PCIe optimizations and dedicated trackers for peer-to-peer PCIe traffic.

Debug Trace Controller (DTC)

The DTC controls distributed *Debug and Trace Monitors* (DTMs) and generates time stamped trace using the ATB interface. DTCs are present in HN-D and HN-T nodes.

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

Configuration Node (CFG)

The CFG is co-located with the HN-D node and handles various CMN-650 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-650
- A configuration bus which connects to all the 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.

Power/Clock Control Block (PCCB)

The 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-650 components, in the following way:

1. The PCCB receives transaction activity indicators from other relevant CMN-650 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-650 components.
3. The PCCB waits for the appropriate responses from the relevant CMN-650 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-650 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 4.1.1 Clock domain configurations on page 63 and 4.2 Power management on page 73.

3.4.4 AMBA 5 CHI to ACE5-Lite bridge (SBSX)

The *AMBA 5 CHI to ACE5-Lite bridge* (SBSX) enables an ACE5-Lite slave device such as a CoreLink™ DMC-400 Dynamic Memory Controller, to be used in a CMN-650 system.

3.4.5 CCIX Gateway (CXG)

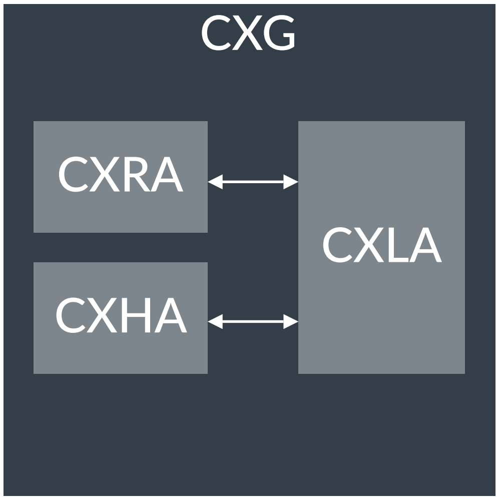
A CXG device bridges between CHI and CXS.

A CXG device contains the following components:

- *CCIX Request Agent* (CXRA) proxy and *CCIX Home Agent* (CXHA) proxy functionality.
- *CXS Link Agent* (CXLA) functionality which is external to the CMN-650 hierarchy.

The following figure shows a CXG block diagram.

Figure 3-6: CXG block diagram



3.4.6 Component Aggregation Layer (CAL)

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

CMN-650 has multiple types of CAL. The different types of CAL support connection of different types of devices.

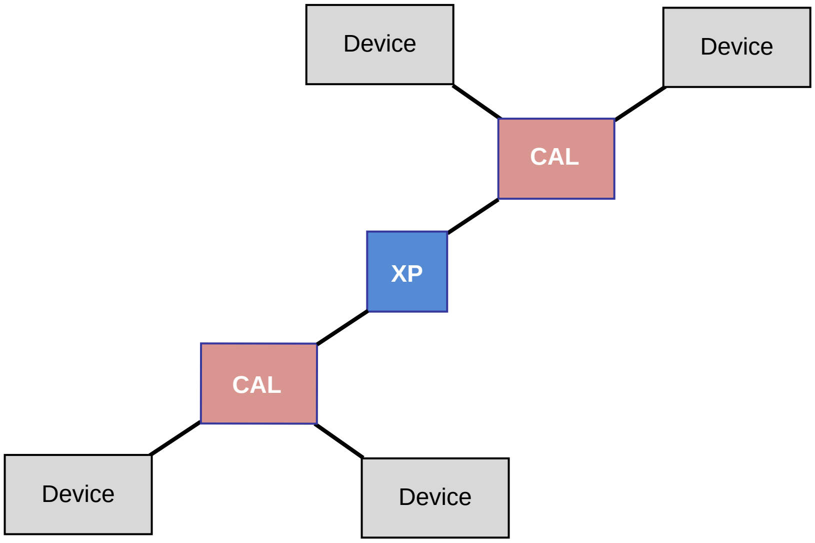
CAL2, CALBYP2
CAL that can connect to two devices. CAL2 supports the following device types:

- RN-F_CHIB_ESAM
- RN-F_CHIC_ESAM
- RN-F_CHID_ESAM
- RN-I
- RN-D
- HN-F
- HN-I
- HN-P
- SBSX

CAL4, CALBYP4
CAL that can connect to four devices. CAL4 only supports the RN-F_CHID_ESAM device type.

All devices that are connected to a single CAL must be of the same type and you must configure them identically. The following figure shows an example XP with two CAL2s.

Figure 3-7: CAL2 example configuration



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 are the variants of each CAL type without bypass functionality. CAL2 and CAL4 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.

3.4.7 Credited Slices (CSs)

You can configure various optional credited register slices in your CMN-650 system. These *Credited Slices* (CSs) can help with timing closure.

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

CMN-650 includes the following optional CSs:

Mesh Credited Slice

Placed between XPs. For more information, see 3.4.7.1 Mesh Credited Slice (MCS) on page 48.

Asynchronous Mesh Credited Slice

Placed between XPs that are in different clock domains. For more information, see 3.4.7.2 Asynchronous Mesh Credited Slice (AMCS) on page 48.

Device Credited Slice

Placed between a device and a CAL, or a device and an XP. For more information, see 3.4.7.3 Device Credited Slice (DCS) on page 50.

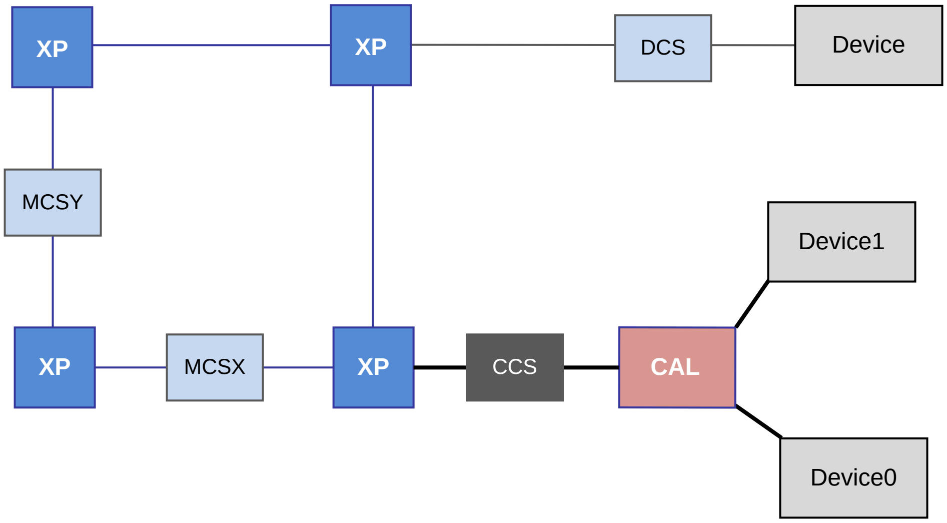
CAL Credited Slice

Placed between a CAL and an XP. For more information, see 3.4.7.4 CAL Credited Slice (CCS) on page 50.

The slices are simple repeater-flop structures that are applied across the entire communication boundary. The supported number of CSs of each type is specified in 3.9 Configurable options for mesh structure on page 56.

The following figure shows where various CSs fit in the structure of the mesh. The example mesh includes two MCSs, which are denoted as MCSX and MCSY according to the X or Y direction of the link. It also includes a DCS and a CCS.

Figure 3-8: Example MCSX, MCSY, CCS, and DCS configuration



3.4.7.1 Mesh Credited Slice (MCS)

You can configure one or more *Mesh Credited Slices* (MCSs) between CMN-650 XPs. MCSs are optional register slices that can help timing closure in a CMN-650 system.

The CMN-650 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-650 implementations to run at higher frequencies.

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

An MCS that is placed between adjacent XPs in the same row is called an MCSX. Similarly, an MCS that is placed between adjacent XPs in the same column is called an MCSY.

3.4.7.2 Asynchronous Mesh Credited Slice (AMCS)

CMN-650 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.

Note

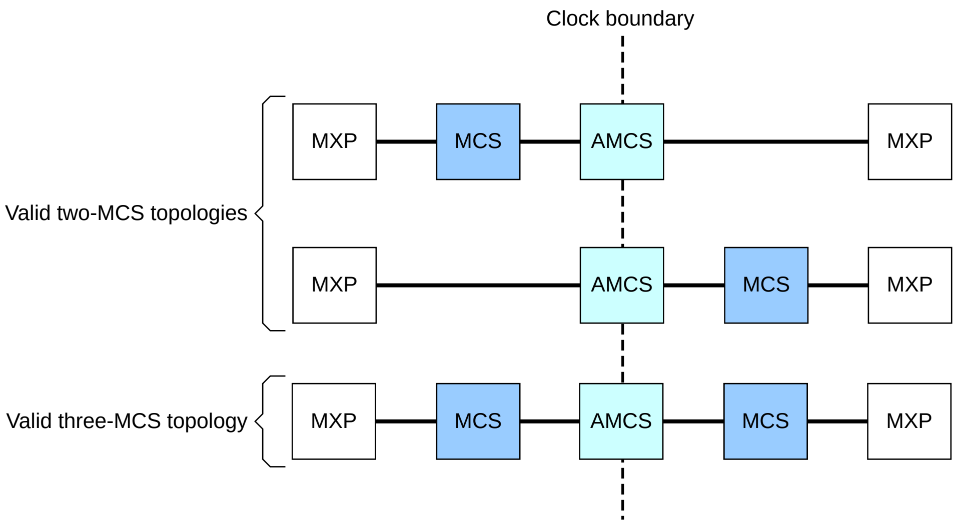
The AMCS does not perform frequency scaling, it only synchronizes traffic between asynchronous clock domains. All CMN-650 clock domains must operate at the same frequency.

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 on CMN-650 support for asynchronous clock domains, see 4.1.1 Clock domain configurations on page 63.

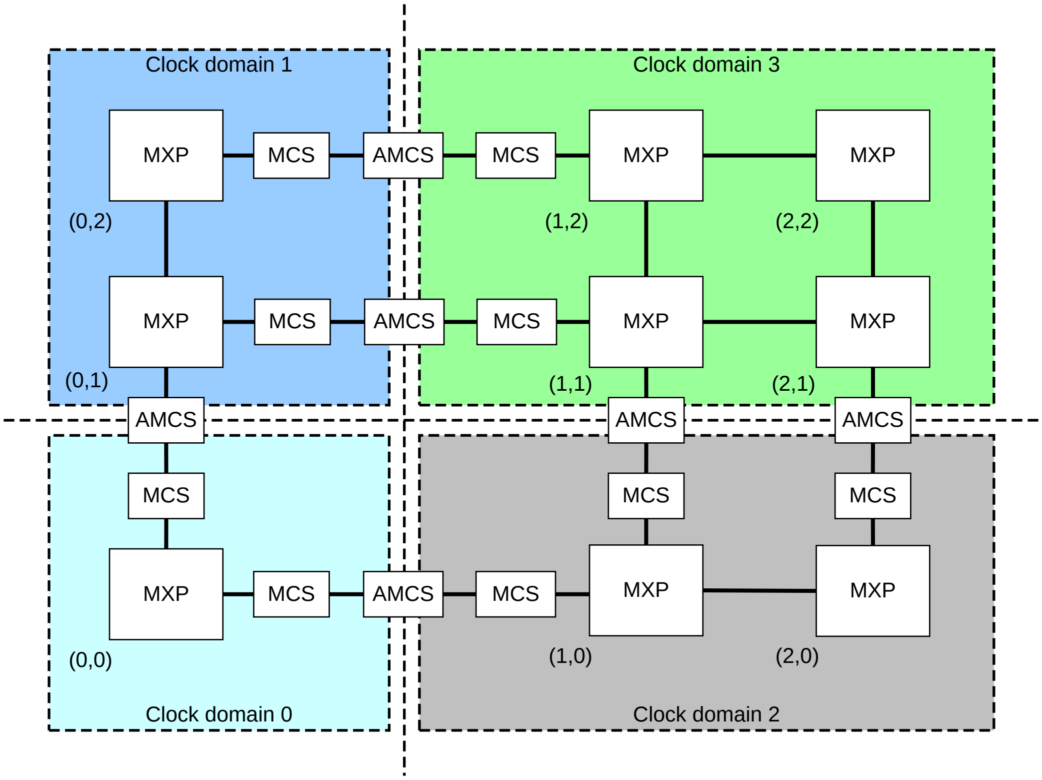
You must configure one or two surrounding MCSs for each AMCS on a link. The following figure shows the supported AMCS plus MCS topologies:

Figure 3-9: Valid AMCS topologies



The following figure shows an example CMN-650 topology with four asynchronous clock domains that are bridged by AMCSs. The clock domains must be enumerated in the order that is shown in the following figure. Clock domain 0 must be the bottom-left quadrant.

Figure 3-10: Example asynchronous mesh topology



3.4.7.3 Device Credited Slice (DCS)

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-650 system.

DCSs are optional register slices that you can add to your CMN-650 configuration. You can add up to four DCSs on any link between a device and an XP.

3.4.7.4 CAL Credited Slice (CCS)

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

3.5 Configure CMN-650

Using Socrates™, you can configure the structure and properties of the CMN-650 to suit the requirements of your system. There are specific steps you can follow to ensure that you have the correct components and configuration as you build and refine your CMN-650 topology.

Before you begin

You must install and familiarize yourself with the Socrates™ IP Tooling platform before configuring CMN-650.
We recommend that you follow the tutorial in *Chapter 4* in the *Arm® Socrates™ for Neoverse™ CMN-650 User Guide* before creating your own system.

About this task

CMN-650 has configuration parameters to customize the properties of the whole mesh and individual devices. You use Socrates™ to configure the values of these parameters and adjust the structure and of your mesh configuration and the placement of devices within it.

For more information about using Socrates™ to configure CMN-650, see the *Arm® Socrates™ for Neoverse™ CMN-650 User Guide*. The *User Guide* is bundled with the Socrates™ tool.

For example system configurations, see 3.10 Example system configurations on page 56 and 3.11 Example CML system configurations on page 60.

Procedure

1. Select your system devices.
   This step identifies the external interfaces and therefore the external and internal devices to specify in the mesh topology. It also identifies some of the internal devices that you require to set up specific CMN-650 functionality. For more information, see 3.6 System component selection on page 51.
2. Configure the mesh size and top-level parameters.
   During this step, you set physical parameters to determine the dimensions of the mesh, and global interconnect configuration parameters. For more information about the constraints on mesh sizing and the global configuration parameters and their values, see the following sections:
   - 3.7 Deciding on the size of the mesh on page 54
   - 3.8 Permitted numbers of devices and system resources in the mesh on page 55
   - 2.4 Global configuration parameters on page 23
3. Place and configure individual devices and CSs within the mesh.
   Each device has its own set of configuration parameters which adjust its behavior. You can also place different types of CSs on certain paths to help with timing closure.
   For more information about the device configuration parameters, see 2.5 Device-level configuration parameters on page 25.

3.6 System component selection

Your system architecture and requirements for certain functionality determine the number and type of components to specify in the mesh topology. Specific components perform specific functions, so requirements differ between CMN-650 configurations.

To understand what components you must specify in your mesh topology, you must consider two factors:

- The functional requirements of your system
- The types of master and slave devices that you have

These factors determine the number and type of external interfaces and the amount of system resources that you need. Therefore, they affect the number and type of devices that you need. The considerations can be split into several categories:

- The number and type of requesting masters in your system
- The number and type of *Home Nodes* (HNs) that you need, which is affected by the following factors:
    - The coherent memory requirements of your system
    - The number and type of I/O slaves or I/O subsystems in your system
- The number of CCIX devices that you have in your system
- The number and type of memory controllers in your system

There are constraints on the number of certain types of components. For the full list of these design constraints, see 3.8 Permitted numbers of devices and system resources in the mesh on page 55.

Requesting masters

Requesting masters are located outside of the mesh and connect to slave ports on CMN-650.

Requesting CHI masters with coherent caches, such as processors, GPUs, or processing elements with internal coherent caches, are referred to as RN-F devices. They connect directly to the CMN-650 interconnect mesh using a CHI RN-F port.

I/O coherent requesting masters are either I/O masters, processing elements without internal caches, or processing elements with internal caches that are not hardware coherent. They connect to CMN-650 RN-I bridge devices using ACE-Lite ports. Each RN-I bridge device has three ACE-Lite interfaces.

A single I/O coherent requesting master can be connected directly to a CMN-650 ACE-Lite port. Alternatively, multiple masters can share a single ACE-Lite port by connecting through external AMBA interconnect components. The following factors affect the number of RN-Is that you need:

- How many of the ports on an RN-I are in use
- How many I/O coherent requesting masters are connected to a single port

Optimal use of RN-Is and the slave interfaces depend on traffic bandwidth requirements and physical floorplan tradeoffs. These constraints are outside the scope of this TRM.

Home Nodes

In a CHI interconnect, you assign a single HN to each byte of the system address space. That HN is responsible for handling all memory transactions that are associated with that address. There are two types of HNs in CMN-650: HN-Fs and HN-Is.

HN-F devices are the HNs for all coherent memory, although they also support non-coherent memory accesses. Memory that is mapped to an HN-F targets DRAM.

Your SLC requirements affect the number of HN-Fs that you need. If your CMN-650 instance has an SLC memory system, each HN-F contains a slice of the SLC. You can configure the size of the SLC slice per HN-F up to a maximum of 4MB. Therefore, the total amount of SLC in your system and the relative size of each SLC slice affects the number of HN-Fs that you need.

Note

The amount of SLC and number of HN-Fs are configured separately.

The other factor that affects the number of HN-Fs that you need is system performance. If your system has many RN-Fs, system performance might be improved by increasing the number of HN-Fs. By increasing the number of HN-Fs, you reduce the amount of traffic targeting each HN-F. However, you must balance this decision with the power and area requirements of your CMN-650 instance.

HN-I devices are the HNs for all memory that targets an ACE-Lite slave device or subsystem. HN-Is do not support coherent memory. However, cacheable transactions can be sent to HN-I.

Each HN-I instance contains a single ACE-Lite master port to send bus transactions to one or more slaves through an AMBA® interconnect. The total ACE-Lite master bandwidth requirement and the physical placement of slave peripherals determines the number of HN-I instances that are needed.

CMN-650 also has the following variants of HN-I that have extra functionality:

HN-T HN-I that has a DTC.
CMN-650 can have zero or more HN-I and HN-T instances.

HN-D HN-I that has a DTC, DVM node, and configuration slave.
CMN-650 must have exactly one HN-D instance.

HN-P HN-I that is optimized for peer-to-peer PCIe traffic.
CMN-650 can have zero or more HN-P instances.

Whether you use a standard HN-I or one of the variants depends on the requirements of your system. The exception to this choice is the HN-D. There must be exactly one HN-D in any CMN-650 instance.

CCIX devices and CML interfaces

The CMN-650 interconnect supports up to ten CXG blocks, each with a CXS interface for connecting external CCIX devices to. The number of CXGs you need in your mesh configuration depends on the following factors:

- The number of CCIX devices in your system
- Whether you group CXGs into *CCIX Port Aggregation Groups* (CPAGs). For more information about CPAGs, see 4.11.6 CCIX Port Aggregation Groups on page 162.

Each CXG contains CXRA, CXHA, and CXLA components. Only the CXRA and CXHA are internal to the CMN-650 hierarchy and function as CCIX RA and HA proxies. The CXLA is external.

Memory controllers

CMN-650 has two types of memory interfaces for connecting different kinds of memory controllers to.

Native CHI memory controllers, such as the CoreLink™ DMC-620 Dynamic Memory Controller, are referred to as SN-Fs. SN-Fs connect directly to CMN-650 through SN-F ports. CMN-650 supports SN-Fs that comply with the following protocols:

- CHI Issue C
- CHI Issue D

You can also connect AXI or ACE-Lite memory controllers to CMN-650 through an ACE-Lite memory interface on an SBSX node. SBSXs bridge between CHI and ACE-Lite protocols. Each SBSX has a single ACE-Lite interface.

The number and type of memory interfaces that you need depends on the design of your system.

Related information

- 2.5 Device-level configuration parameters on page 25

3.7 Deciding on the size of the mesh

The number of XPs and their arrangement determines the size of the mesh. The number of XPs that you require depends on the number of devices in your configuration. There are also constraints on how the XPs can be arranged.

You require one XP for every two devices in your configuration. Therefore the minimum number of XPs that you require in your configuration is half of the number of devices, rounded up.

The mesh must be rectangular, so you might require more than the minimum number of XPs to complete the mesh. For example, consider a configuration with 13 devices, which requires a minimum of seven XPs. In this case, to ensure that the mesh is rectangular, the configuration must contain 8 XPs. The XPs in this example could be arranged in a 2×4 or 4×2 mesh.

Related information

- 3.8 Permitted numbers of devices and system resources in the mesh on page 55
- 3.9 Configurable options for mesh structure on page 56

3.8 Permitted numbers of devices and system resources in the mesh

CMN-650 has constraints on the amount of some system resources and the number of each device type that you can use in the configuration.

The following table shows the permitted amounts of system resources and numbers of devices in CMN-650.

Table 3-5: Amount of system resources and number of devices

Category	Resource	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. RN-Fs can be one of the following three types: • RNF_CHIB_ESAM • RNF_CHIC_ESAM • RNF_CHID_ESAM	1-256 without CAL 2-256 with CAL	In a CML system, the maximum number of RN-Fs across all chips is 512. For more information, see 4.3.3 Extended CCIX Requesting Agent ID mechanism for up to 512 RN-Fs on page 92.
I/O resources	Number of RN-Is	The number of RN-I instances in the system	0-36 2-36 with CAL	At least one RN-I or RN-D must be present. The total count of RN-Is and RN-Ds must not exceed 36.
	Number of RN-Ds	The number of RN-D instances in the system	0-36 2-36 with CAL	
	Number of HN-Is	The number of HN-I instances in the system. This count includes HN-T, HN-P, and the HN-D which is always present.	1-16 2-16 with CAL	CMN-650 supports connection of HN-I and HN-P nodes to CAL. You cannot connect HN-D and HN-T nodes to CAL. For meshes where the total number of XP and CXG instances is greater than 63, at least one HN-T node is required in the mesh.
Debug resources	Number of DTCs	The total number of DTC domains	1-4	The number of DTCs equals the number of HN-T nodes plus the HN-D node.
SLC	Number of HN-Fs	The total number of HN-F instances in the system	1-64	-
Memory resources	Number of SN-Fs	The number of SN-Fs (CHI interfaces)	0-40	At least one SN-F or SBSX must be configured. The total number of SN-Fs and SBSXs must not exceed 40. CMN-650 supports connection of SBSXs to CAL.
	Number of SBSXs	The number of SBSX instances (AXI interfaces)	0-40	

Category	Resource	Description	Values (default)	Comments
CML resources	Number of CXGs	The number of CXG instances	0-10	-

3.9 Configurable options for mesh structure

These configurable options define the dimensions of the mesh and the number of different types of CSs across different links. Each value has a specific range and there are some constraints that apply to these values.

The following table shows the configurable options for mesh structure.

Table 3-6: Configurable options for mesh structure

Parameter	Description	Values	Comments
Mesh X dimension	Number of mesh columns	1-10	The following mesh configurations are not supported: • 1 × 2 • 2 × 1
Mesh Y dimension	Number of mesh rows	1-10	
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	0-2	

Related information

- 2.4 Global configuration parameters on page 23
- 2.5 Device-level configuration parameters on page 25

3.10 Example system configurations

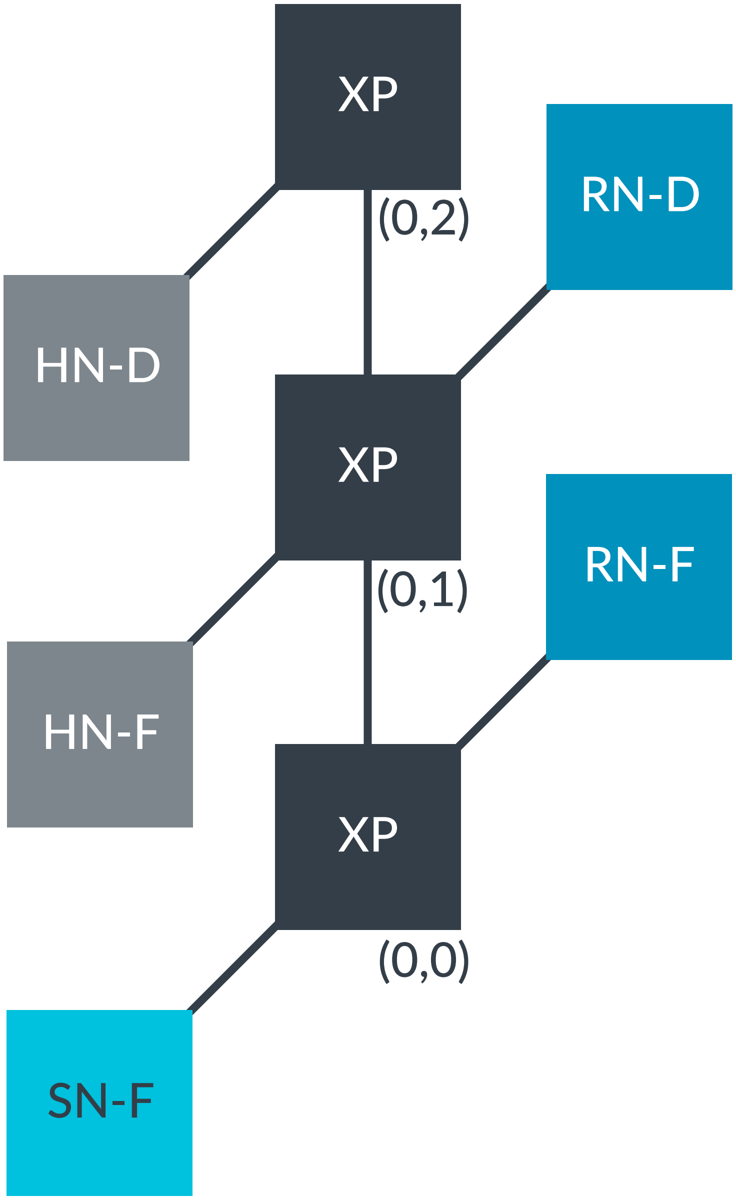
Because CMN-650 is highly configurable, different system configurations can vary significantly. These examples show possible small, medium, and large configurations, with varying node types and system components.

The following figure shows a small 1 × 3 mesh configuration with single instances of the following node types:

- RN-F
- HN-F
- RN-D

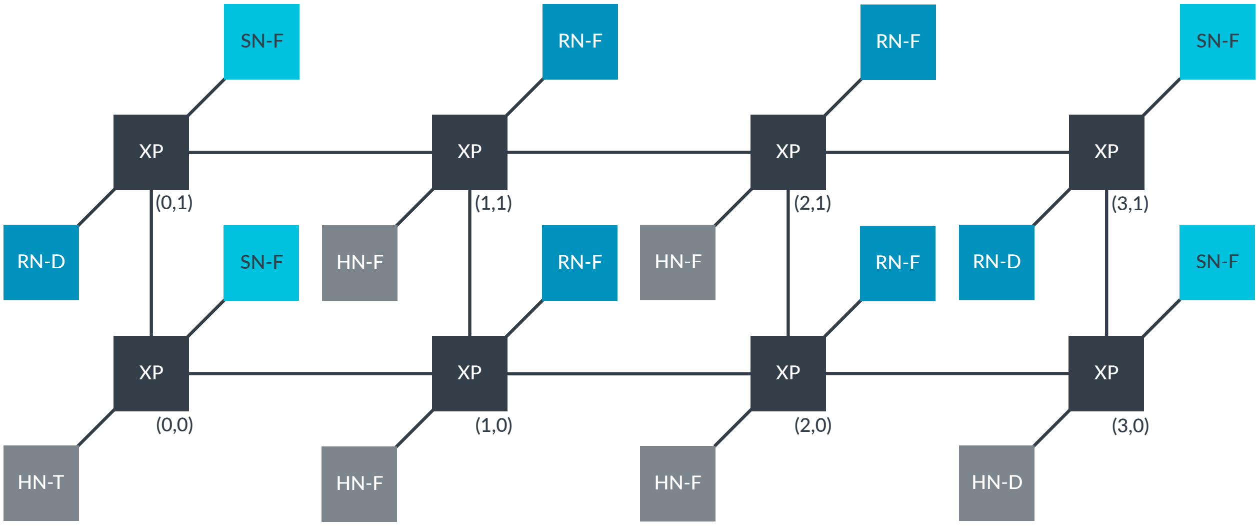
- SN-F
- HN-D

Figure 3-11: Example 1 × 3 mesh configuration



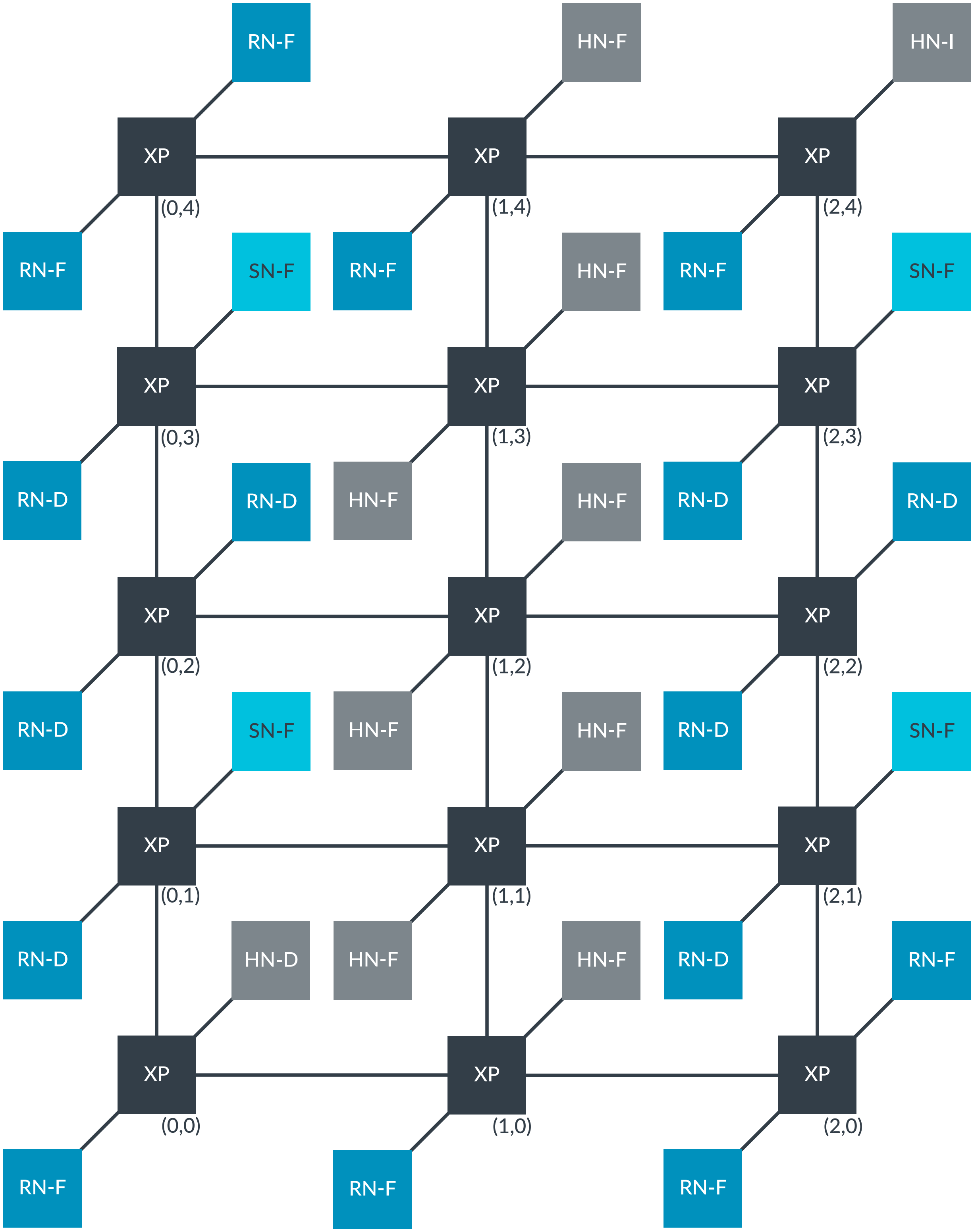
The following figure shows a medium 4 × 2 mesh configuration with single and multiple instances of the node types in the preceding figure.

Figure 3-12: Example 4 × 2 mesh configuration



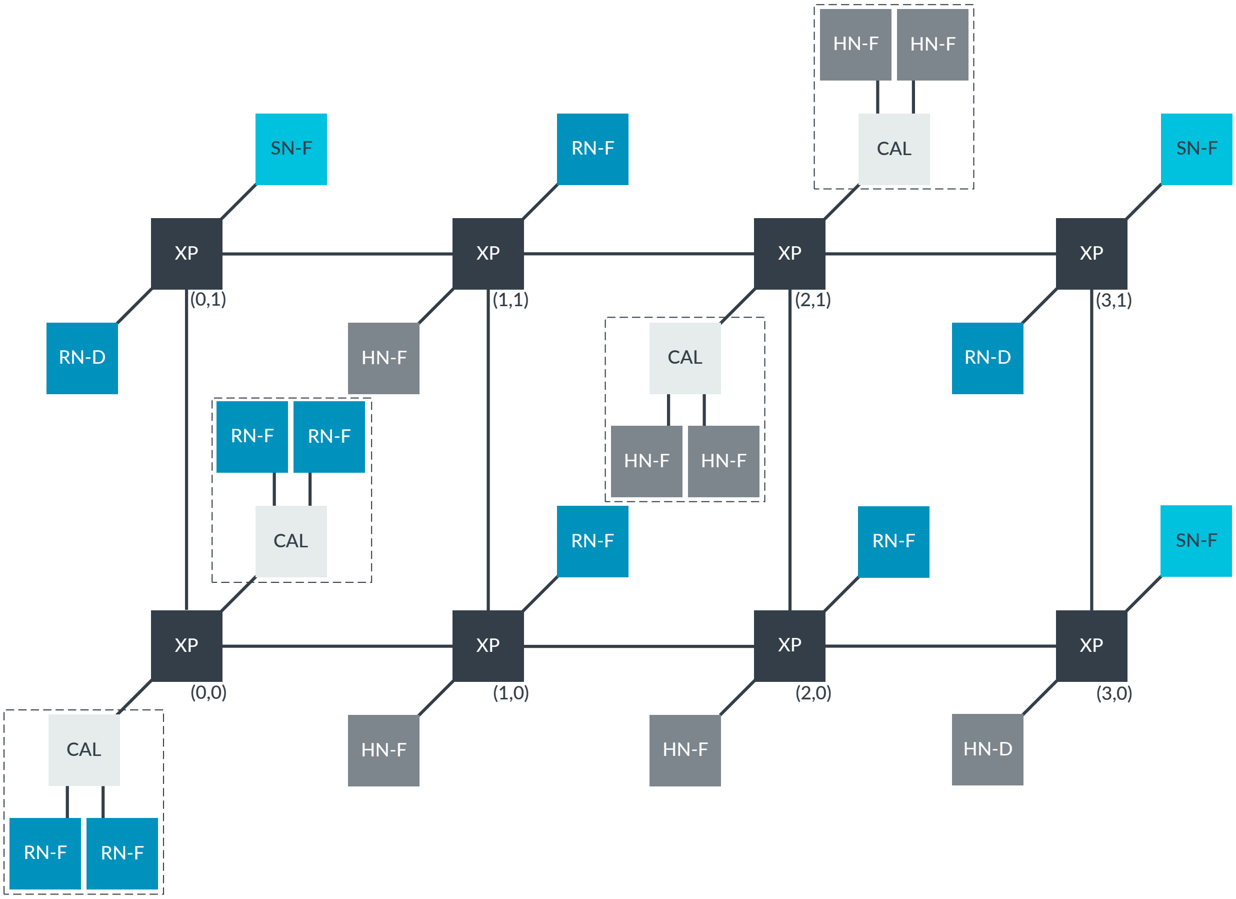
The following figure shows a large 3 × 5 mesh configuration with the same node types as the other examples.

Figure 3-13: Example 3 × 5 mesh configuration



The following figure shows a medium 4 × 2 mesh configuration with RN-F and HN-F CAL, shown by the dotted lines.

Figure 3-14: Example 4 × 2 mesh configuration with CAL



3.11 Example CML system configurations

You can use CMN-650 CXG nodes to create CML systems with other CCIX-compatible hardware. The topologies you can create using the CXG block are varied. These examples show CCIX topologies with single shared ports, multiple shared ports, and connectivity using external PCIe components.

The following figures show three simplified CCIX topology examples.

Figure 3-15: Two-socket CCIX system with a single CCIX port connection

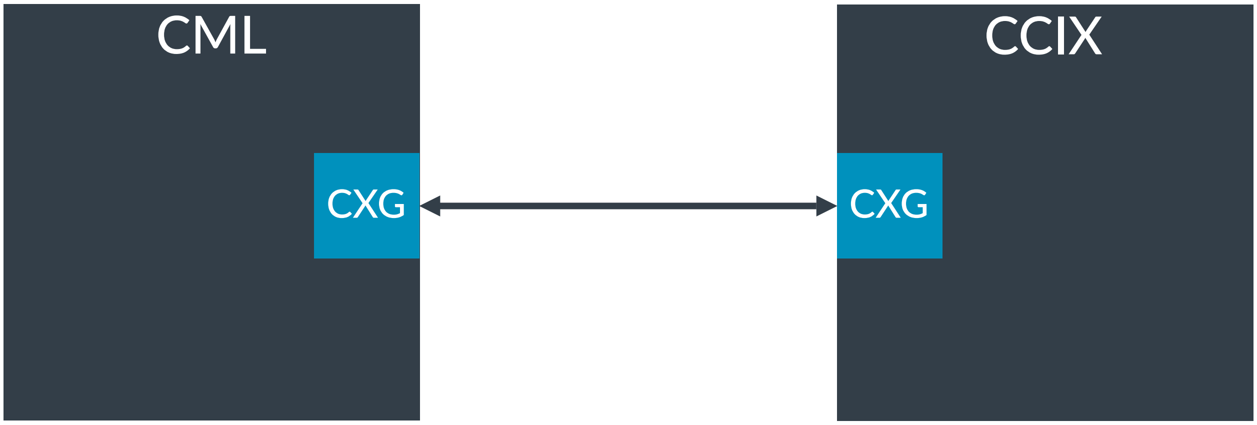


Figure 3-16: Two-socket CCIX system with CPA

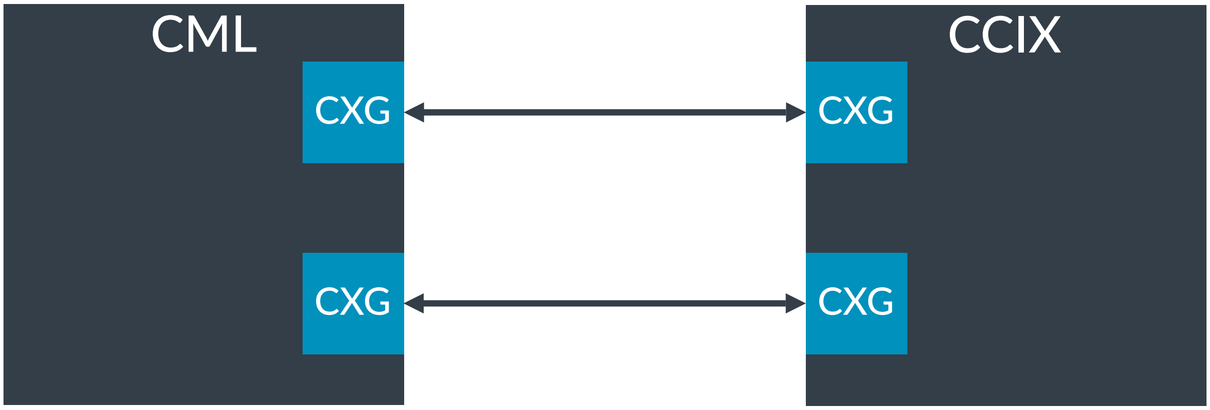
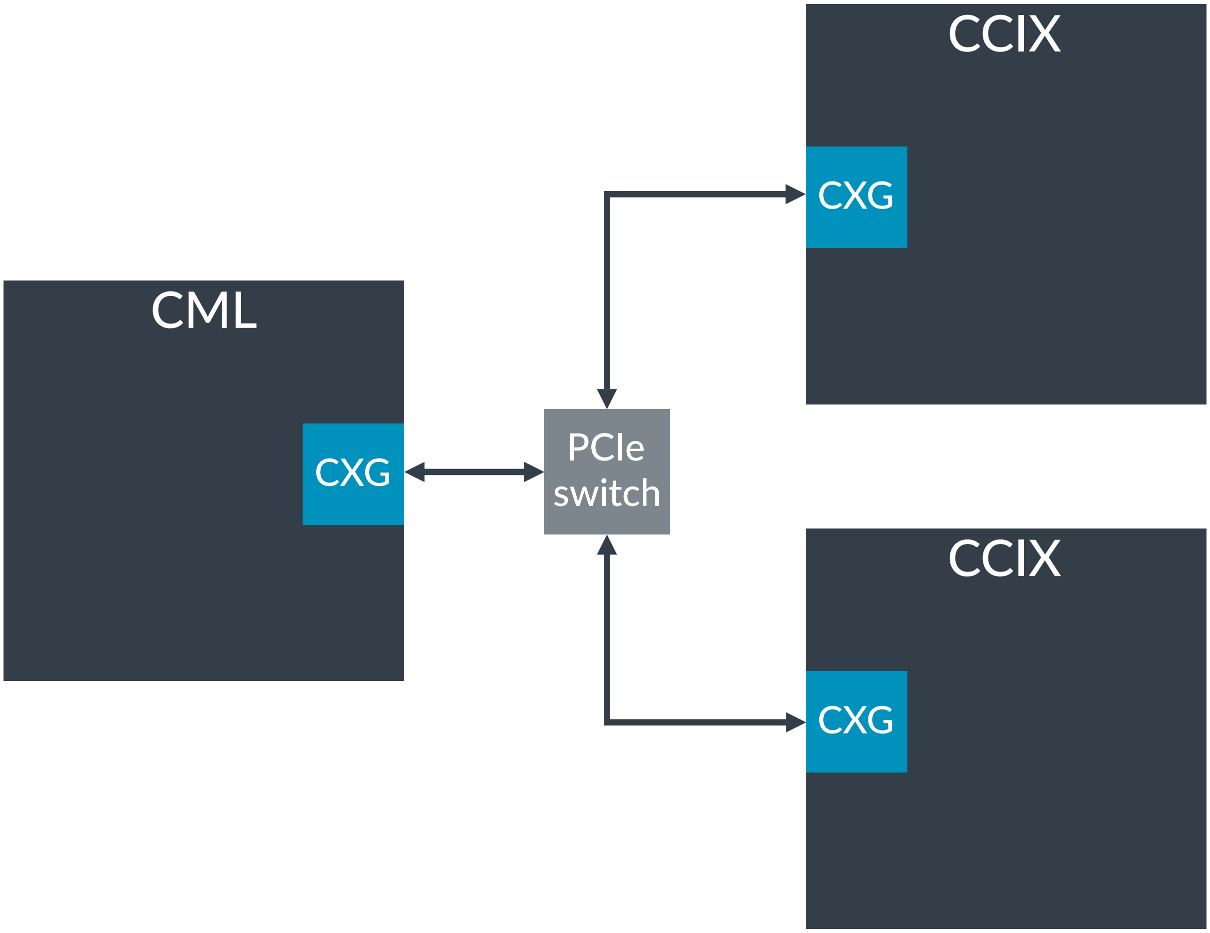


Figure 3-17: Three-socket CCIX system with a PCIe switch



You can create more switched topologies by enabling port-to-port forwarding. For more information, see 4.11.11 CML port-to-port forwarding on page 165.

4 Functional description

This chapter describes functionality achieved when you design and configure the CMN-650 interconnect and its components.

4.1 Clocks and resets

CMN-650 provides a hierarchical clocking microarchitecture which enables dynamic clock management for power efficiency. It also has a global reset signal.

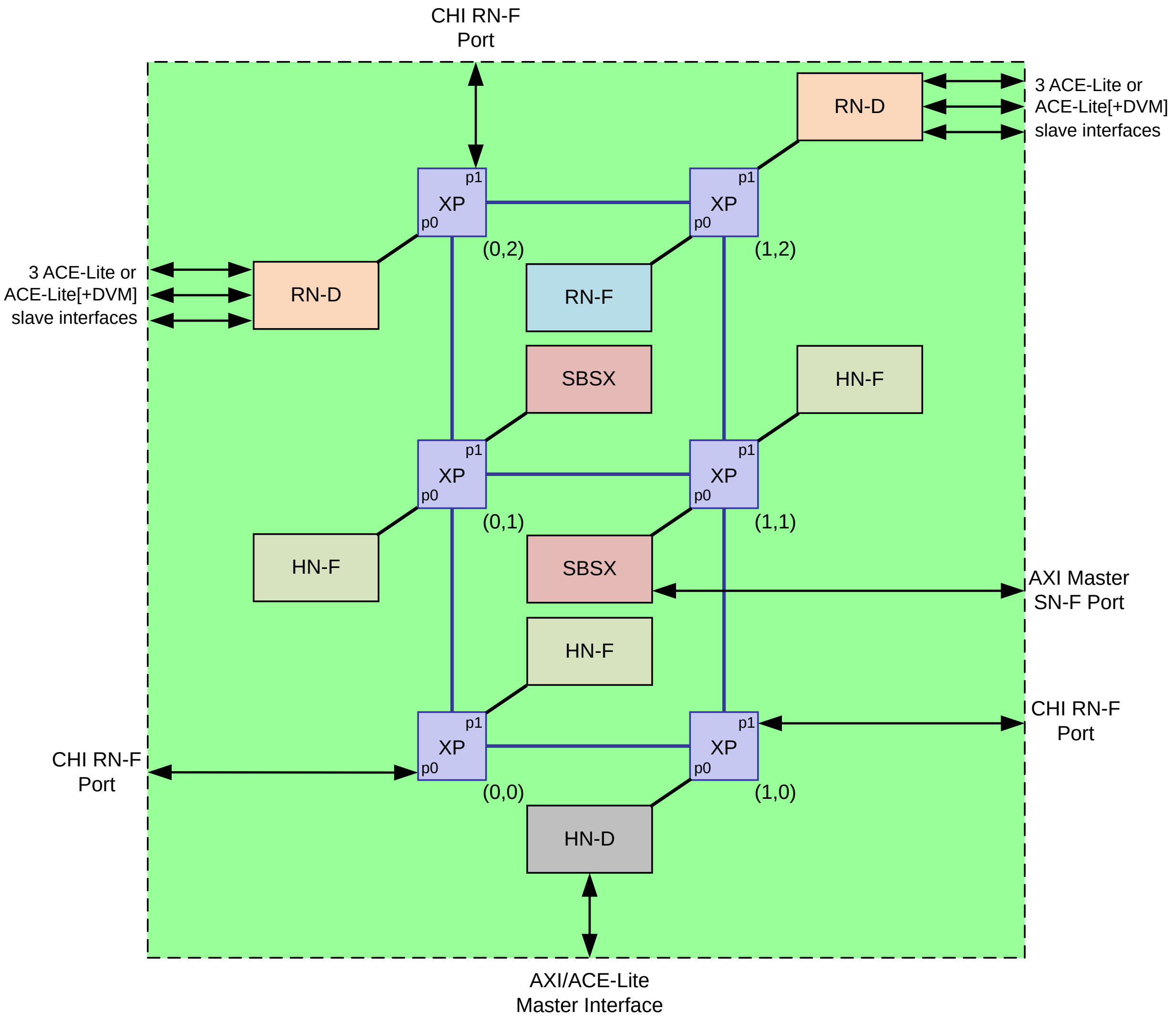
CMN-650 has a *High-level Clock Gating* (HCG) mechanism for clock signal management during periods of inactivity. For more information, see 4.1.5 High-level Clock Gating (HCG) on page 69.

4.1.1 Clock domain configurations

CMN-650 supports either one synchronous or multiple asynchronous clock domains.

The following figure shows a CMN-650 configuration operating in a single and fully synchronous clock domain.

Figure 4-1: CMN-650 topology with fully synchronous clock domain

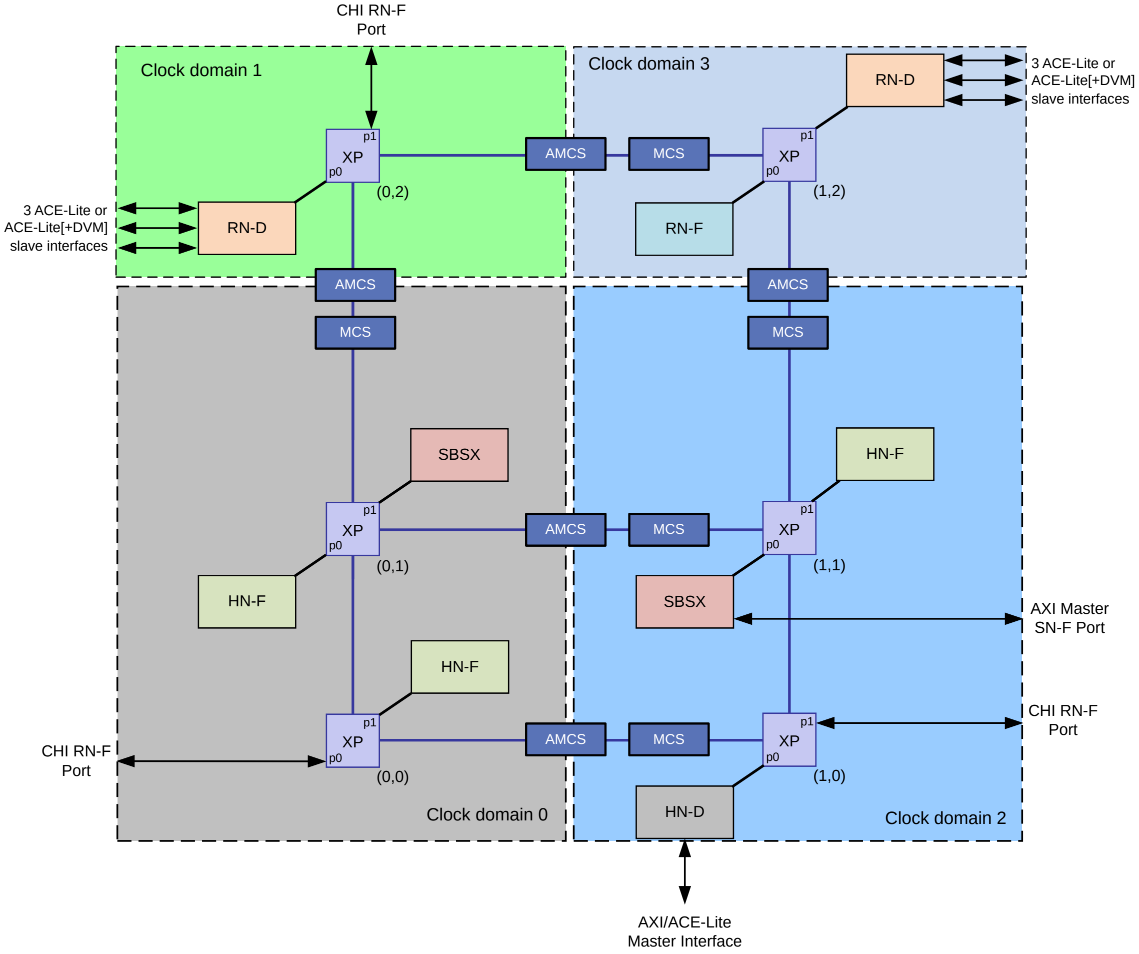


The global clock signal in a fully synchronous clock domain configuration is known as **GCLK0**.

For larger mesh topologies, synchronization and clock skew can be problematic because of the large distances that clock signals travel. Therefore, CMN-650 also supports dividing the mesh into four asynchronous clock domains. This feature is configured using Socrates™.

The following figure shows an example CMN-650 configuration with four asynchronous clock domains.

Figure 4-2: CMN-650 topology with two 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-650 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.
- You must configure the AMCS that bridges the two clock domains next to one or two MCSs in a chain. For more information about the allowed topologies, see 3.4.7.2 Asynchronous Mesh Credited Slice (AMCS) on page 48.
- The individual clock signals that supply each clock domain must run at the same frequency as each other, although they can be asynchronous.
- CMN-650 only supports rectangular clock domains containing one or more XPs. L-shaped or other clock domain topologies are not supported.
- 2x2 and smaller mesh configurations do not support multiple clock domains.

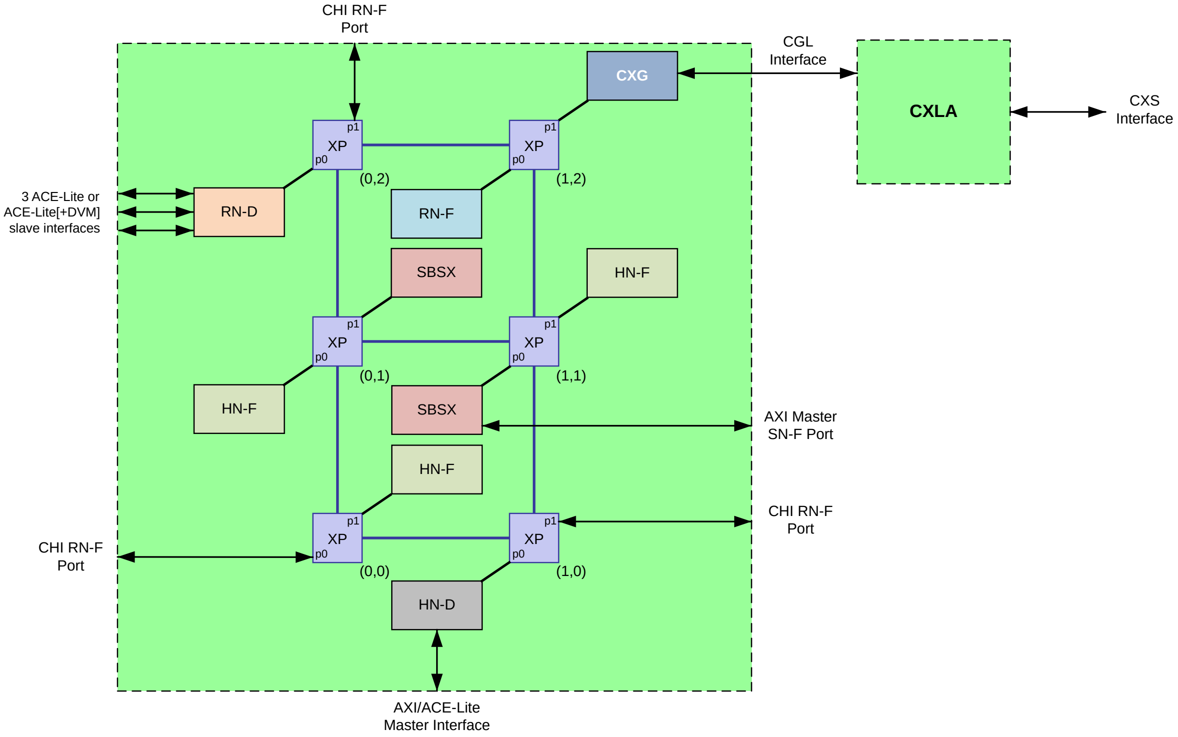
- A single Q-Channel controls all the clock signals. Therefore the clock domains cannot be gated separately, and must be powered up and down in the same way as a single synchronous clock domain.
- The mesh can only have a single power domain.

4.1.2 CML clock inputs

There are two extra clock inputs for CML configurations: **CLK_CGL** and **CLK_CXS**.

CLK_CGL is a copy of the CMN clock input used by the CXRH node associated with the CXLA, **GCLKn**. A separate clock input is provided to allow gating of the CGL clock domain independent of the **GCLKn** domain. **CLK_CXS** clocks the CXS interface logic, and can be synchronous or asynchronous to **GCLKn**. **CLK_CXS** can be driven with **CLK_CGL** for synchronous configurations, as the following figure shows.

Figure 4-3: CMN-650 clock domains with synchronous CXS domain

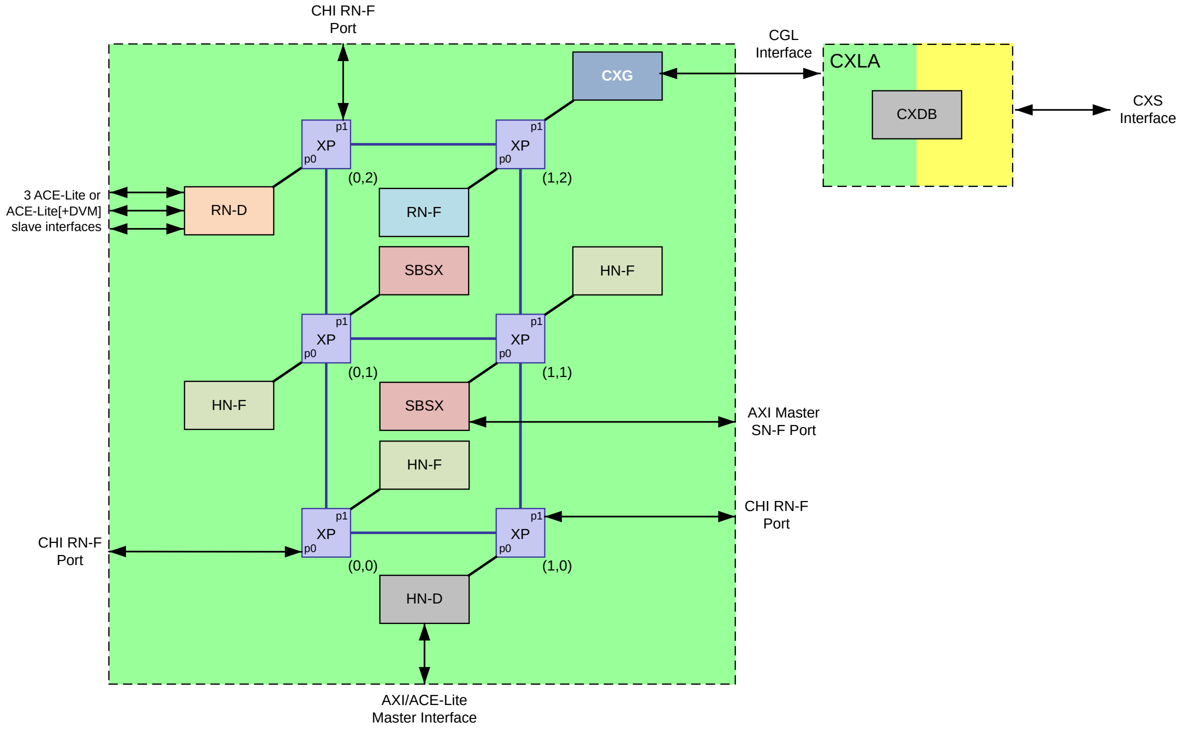


Note

If the CXS interface is asynchronous to **GCLKn**, **GCLKn** clocks the corresponding CXRH logic. In other words, the global clock signal for the clock domain that the CXRH sits in clocks the CXRH.

The CXLA block contains an asynchronous domain bridge for configurations where the **CLK_CXS** domain is asynchronous to the **CLK_CGL** domain, as the following figure shows.

Figure 4-4: CMN-650 clock domains with asynchronous CXS domain



4.1.3 Clock hierarchy

The clocking delivery and clock gating architecture are hierarchical.

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

Global clock
The global clock is the clock input to the CMN-650 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-650 includes support for external clock control.

Note

If you configure CMN-650 to use multiple asynchronous clock domains, a single Q-Channel controls each individual clock signal. Therefore, the individual clock domains cannot be separately gated. For more information, see 4.1.1 Clock domain configurations on page 63.

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-650 RTL. The exact set of regional clocks is internal to CMN-650 and is not described in this book.

Local clocks

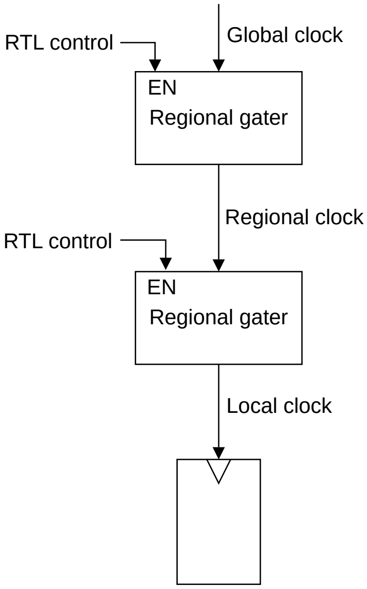
Local clocks are created according to the following hierarchy:

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-650. The exact set of local clocks is internal to CMN-650 and is not described in this manual.

The following figure shows the clocking hierarchy.

Figure 4-5: Clocking hierarchy



4.1.4 Clock enable inputs

CMN-650 includes several clock enable inputs, enabling synchronous communication with slower SoC logic.

Note

The following description provides 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:

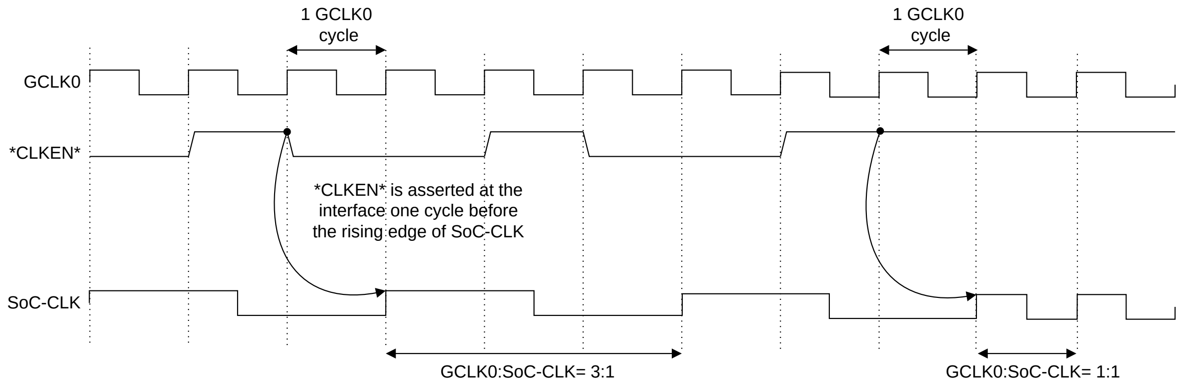
ACLKEN_S	This input is present on each AMBA® slave interface.
ACLKEN_M	This input is present on each AMBA® master interface.
ATCLKEN	This input is present on each debug and trace ATB interface.

All clock enable signals, shown here as ***CLKEN***, have identical functionality: they enable their respective interface to run at integer fractions of **GCLK0**. In other words, the clock enable signals run slower than **GCLK0**. **ACLKEN_S** and **ACLKEN_M** can run at any integer ratio from 1:1 to 4:1 of the frequency of **GCLK0**. **ATCLKEN** is limited to 1:1, 2:1, and 4:1 integer ratios.

CLKEN asserts one **GCLK0** cycle before the rising edge of the SoC clock, **SoC-CLK**. SoC control logic can change the ratio of **GCLK0** frequency to **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 respective to **GCLK0** from 3:1 to 1:1.

Figure 4-6: *CLKEN* with GCLK0:SoC-CLK ratio changing from 3:1 to 1:1



4.1.5 *High-level Clock Gating* (HCG)

The PCCB supports a *High-level Clock Gating* (HCG) mechanism. This mechanism notifies the SoC when CMN-650 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 4.1.6 External Clock Controller (ExtCC) on page 70.

Note

If your CMN-650 configuration has multiple asynchronous clock domains, the ExtCC stops each individual global clock signal identically.

CMN-650 includes a Q-Channel interface that enables CMN-650 and the SoC to communicate to achieve HCG functionality through the PCCB. For more information, see the *AMBA® Low Power Interface Specification Arm® Q-Channel and P-Channel Interfaces*.

4.1.6 *External Clock Controller* (ExtCC)

The *External Clock Controller* (ExtCC) is used to control the HCG 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:

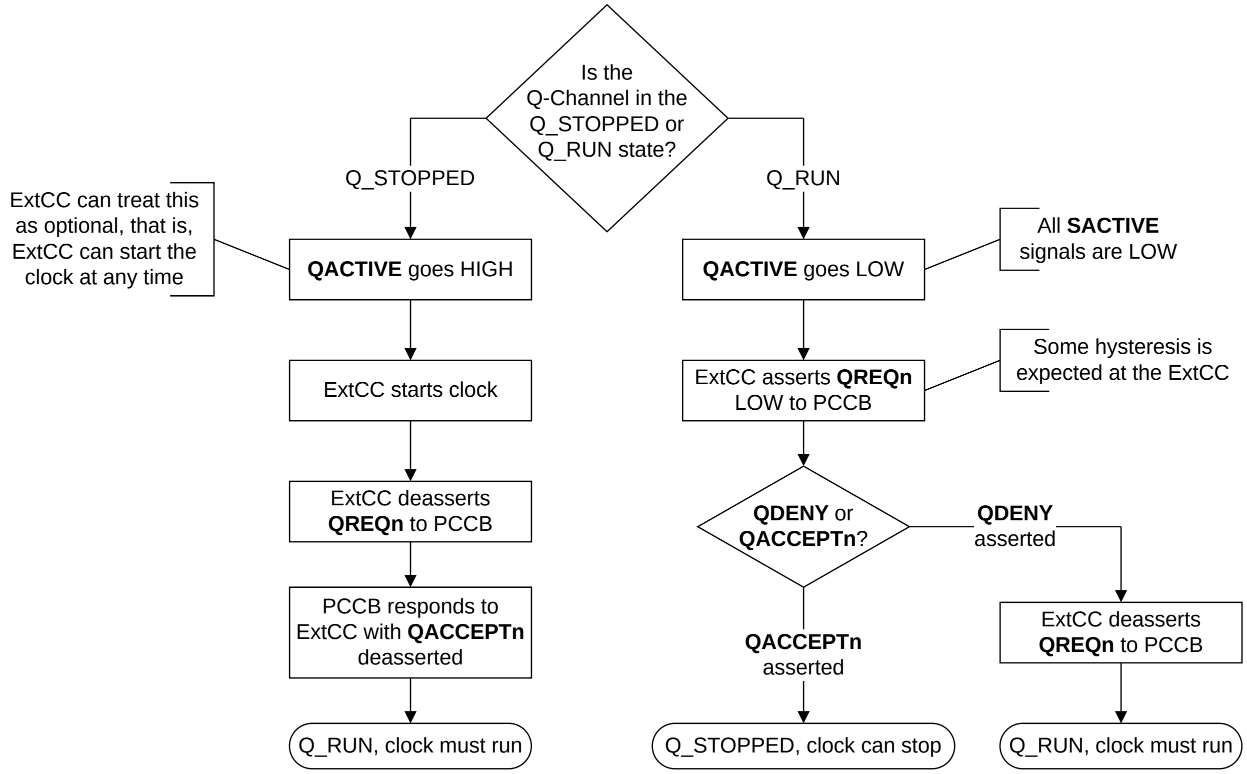
Q_STOPPED

Quiescent state, where **QREQn** and **QACCEPTn** are asserted.

Q_RUN

Active state, where **QREQn** and **QACCEPTn** are deasserted.

Figure 4-7: Clock gating control using ExtCC


Is the Q-Channel in the Q_STOPPED or Q_RUN state?
Q_STOPPED
Q_RUN
ExtCC can treat this as optional, that is, ExtCC can start the clock at any time
QACTIVE goes HIGH
QACTIVE goes LOW
All SACTIVE signals are LOW
ExtCC starts clock
ExtCC asserts QREQn LOW to PCCB
Some hysteresis is expected at the ExtCC
ExtCC deasserts QREQn to PCCB
QDENY or QACCEPTn?
QDENY asserted
PCCB responds to ExtCC with QACCEPTn deasserted
QACCEPTn asserted
ExtCC deasserts QREQn to PCCB
Q_RUN, clock must run
Q_STOPPED, clock can stop
Q_RUN, clock must run

The requirements of the ExtCC are as follows:

- It must supply a clock to CMN-650 when the Q-Channel is in any state other than Q_STOPPED.
- The ExtCC can either:
  - Choose to gate the clock to CMN-650 when the Q-Channel is in the Q_STOPPED state.
  - Choose to run the clock at any time.
- ExtCC is responsible for bringing the Q-Channel to Q_RUN state after reset deassertion.
- This manual does not describe the exact behavior of the ExtCC and its usage of **QREQn** in response to **QACTIVE** deassertion. However, 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-650 can be negatively affected.
- It is the responsibility of the SoC designer to fully control the clock management Q-Channel. If you require a control or configuration bit to completely enable or disable HCG functionality, it must exist outside of CMN-650. CMN-650 has no internal means of disabling HCG.

4.1.7 CML clock management

CMN-650 CML configurations add one or two extra clock domains, depending on whether you configure the CXS logic to be synchronous or asynchronous to **GCLKn**.

Note

CLK_CGL can drive **CLK_CXS** in synchronous configurations. For more information, see 4.1.2 CML clock inputs on page 66.

Synchronous or asynchronous configuration is based on the *CXLA_DB_PRESENT* parameter.

CML configurations add Q-Channel interfaces for each CXG instance and corresponding CXS interface:

CLK_CGL Q-Channel

Manages the CGL link and CGL domain logic in the CXG and CXLA devices

CLK_CXS Q-Channel

Manages the CXS link interface and CXS clock domain logic in the CXLA

The following table shows the possible clock states for the CMN-650 and CML device clocks, where N denotes multiple CXS interfaces:

Table 4-1: CMN-650 and CML device clock states

GCLKn	CLK_CGL[N]	CLK_CXS[N]	Description
RUN	RUN	RUN	CMN-650 and CXS[N] interface active
RUN	RUN	STOP	CXS[N] domain gated. This state is only valid for asynchronous configurations where **CLK_CGL** is asynchronous to **CLK_CXS**.
RUN	STOP	RUN	CGL[N] inactive, transitory state. This state is only valid for asynchronous configuration where **CLK_CGL** is asynchronous to **CLK_CXS**.
RUN	STOP	STOP	CXS[N] interface fully gated
STOP	STOP	RUN	CXS[N] active, others inactive, transitory state. This state is only valid for asynchronous configuration where **CLK_CGL** is asynchronous to **CLK_CXS**.
STOP	STOP	STOP	CMN-650 fully gated, all CXS[N] interfaces inactive

Note

If your configuration has multiple asynchronous clock domains, then the information in the preceding table applies equally to each global clock signal.

4.1.8 Reset

CMN-650 has a single global reset input signal, **nSRESET**.

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-650 components enter and exit their reset states correctly.

All CMN-650 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**.

4.1.9 CML reset

There are two extra reset inputs for the CML configuration: **nRESET_CGL** and **nRESET_CXS**.

Both **nRESET_CGL** and **nRESET_CXS** are active-LOW signals that can be asynchronously or synchronously asserted and deasserted.

When asserted, **nRESET_CGL** and **nRESET_CXS** must remain asserted for 20 **CLK_CGL** and **CLK_CXS** clock cycles respectively. Likewise, when deasserted, **nRESET_CGL** and **nRESET_CXS** must remain deasserted for 20 **CLK_CGL** and **CLK_CXS** clock cycles respectively. This requirement ensures that all CML components enter and exit their reset states correctly.

Both **CLK_CGL** and **CLK_CXS** must be active during the required 20-cycle, or larger, period of **nRESET_CGL** and **nRESET_CXS** assertion respectively. **CLK_CGL** and **CLK_CXS** must also remain active for at least 20 cycles following deassertions.

For more relationship information between the CXS and CGL domains, refer to 4.1.2 CML clock inputs on page 66.

Note

There is no sequencing requirement between the CMN-650 **nSRESET** and the CML resets. However, the CML domains must exit reset before CXLA functionality is required.

4.2 Power management

CMN-650 includes several power management capabilities, that are either externally controllable or that the SoC assists.

CMN-650 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 elsewhere. For more information, see 4.1 Clocks and resets on page 63.

4.2.1 Power domains

The power domains in CMN-650 are divided into logic domains and domains for the RAMs in the HN-F partitions.

CMN-650 has the following power domains:

Logic

All logic except HN-F SLC tag and data RAMs and HN-F SF RAMs.

SLC RAM0

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

SLC RAM1

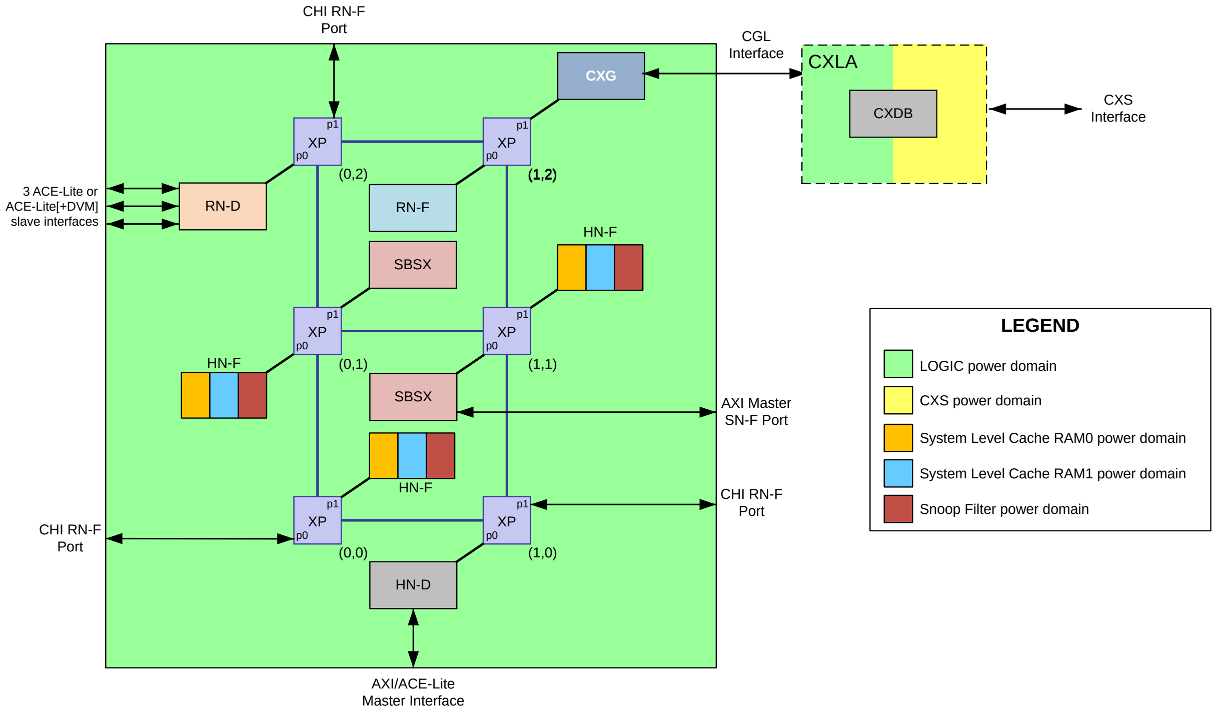
SLC tag and data RAMs, way[15:8] within HN-F partitions. The RAMs in each HN-F partition can be independently controlled. For 3MB SLC size configurations, the RAM1 domain includes way[11:8].

Snoop filter only mode

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

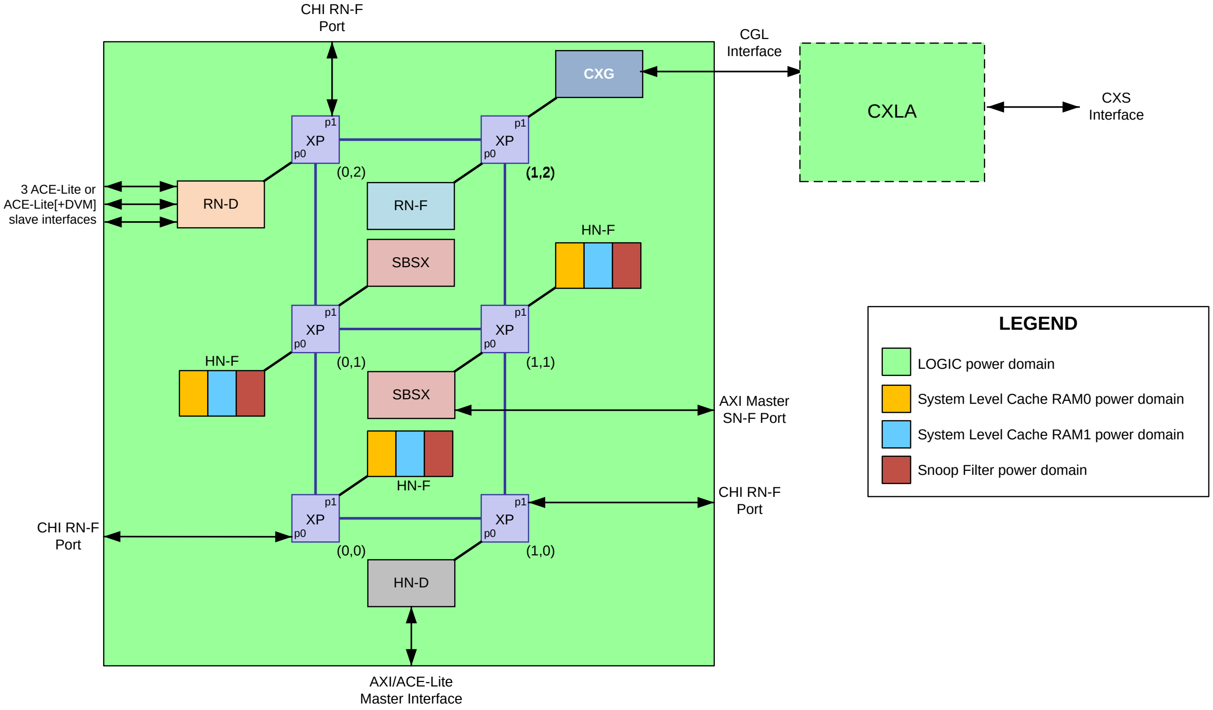
The following figure shows an example power domain configuration.

Figure 4-8: CMN-650 power domain example



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

Figure 4-9: Single CML power domain example

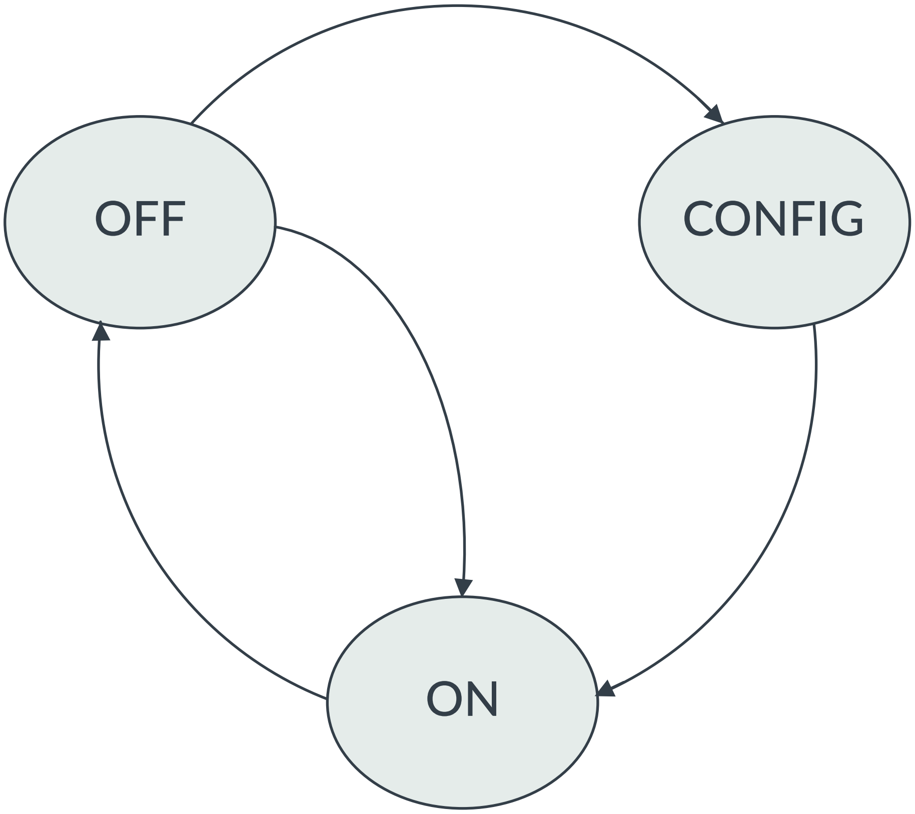


4.2.2 Power domain control

The CMN-650 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 4-10: 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-650 is exiting a memory retention state. The transition 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 4-2: Power mode configurations and PSTATE values

Power mode	PSTATE	CMN-650 logic	HN-F
OFF	`0b00000`	OFF	OFF/MEM_RET
CONFIG	`0b11000`	ON	ANY
ON	`0b01000`	ON	ANY

Note

The CMN-650 P-Channel interface requires clocks when either **QACTIVE_ICN** or **QACTIVE_DEV** are asserted to function properly, including **PACTIVE**.

For an introduction to HN-F states, see 4.2.6 HN-F power domains on page 80.

For P-Channel signal information, see B.11 Power management signals on page 986.

CXS power gating

CML systems contain an extra power domain, the CXS power domain, which is present according to the value of the CXLA `DB_PRESENT` parameter. CXS logic is controlled by a combination of the CXS power Q-Channel and the CXS Q-Channel that controls the clocks. Unless both the CXS power Q-Channel and the CXS Q-Channel are in the OFF state, power must be provided.

4.2.3 P-Channel on device reset

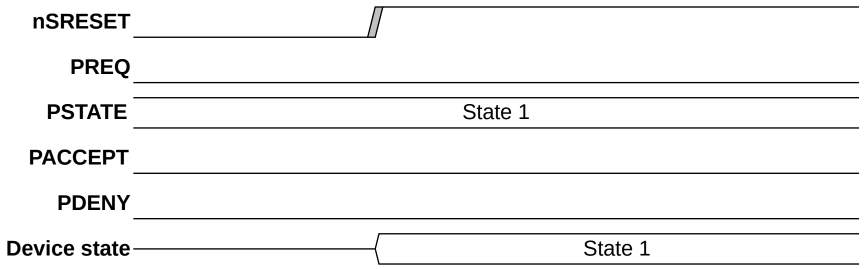
This section shows how to initialize the power state of a power domain.

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-650. 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-650 also supports P-Channel initialization with **PREQ** asserted at **nSRESET** deassertion.

Note

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

Figure 4-11: Reset state initialization with PREQ deasserted



For an introduction to HN-F states, see 4.2.6 HN-F power domains on page 80.

4.2.4 CXS power domain

CML systems contain an extra power domain, the CXS power domain. The CMN-650 CXS power Q-Channel controls the CXS power domain.

The CXS power domain is present depending on the value of the CXLA *DB_PRESENT* parameter.

Note

If the CXS interface is inactive, the CXS power domain can be shut off.

The following table shows the possible CMN-650 LOGIC and CXS power states, where N denotes multiple CXS interfaces:

Table 4-3: CMN-650 LOGIC and CXS power states

LOGIC state	CXS[N] state	Description
ON	ON	CMN-650 and CXS[N] interface active.
ON	OFF	CXS[N] interface inactive.
OFF	ON	CXS[N] domain active, transitory state.
OFF	OFF	Shut down, all CXS[N] interfaces inactive.

4.2.5 HN-F Memory retention mode

When isolating the CMN-650 outputs, handshake protocols on certain interfaces must be followed. This section describes the steps to take 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-650 in the LOGIC_OFF state through the logic P-Channel.
4. Isolate the CMN-650 outputs. If the logic on the other side of the interface is being powered down or reset, this step might not be needed.
5. Turn off power to CMN-650.

Exiting HN-F Memory retention mode

1. Apply power to CMN-650.
2. Assert reset.
3. Enable clocks.

4. Disable isolation of the CMN-650 outputs.
5. Deassert reset.
6. Place CMN-650 in LOGIC_CONFIG state through the logic P-Channel.
7. Reprogram the por_hnf_ppu_pwpr register to the retention mode the HN-F was in before turning off power.
8. Reprogram the por_hnf_ppu_pwpr register to ON.
9. Reprogram the CMN-650 configuration registers, including the RN SAM and any other registers that are written during cold boot.
10. Place CMN-650 in the LOGIC_ON state through the P-Channel.
11. Resume traffic and normal operation.

4.2.6 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 the following 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 operate in four modes:

***Full Associativity Mode* (FAM)**

The SF and the entire SLC are enabled.

***Half Associativity Mode* (HAM)**

The SF is enabled but the upper half of the SLC ways are disabled and powered off.

***Snoop Filter Only Mode* (SFONLY)**

The SF is enabled but the whole SLC is powered off.

***No SLC Mode* (NOSFSLC)**

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.

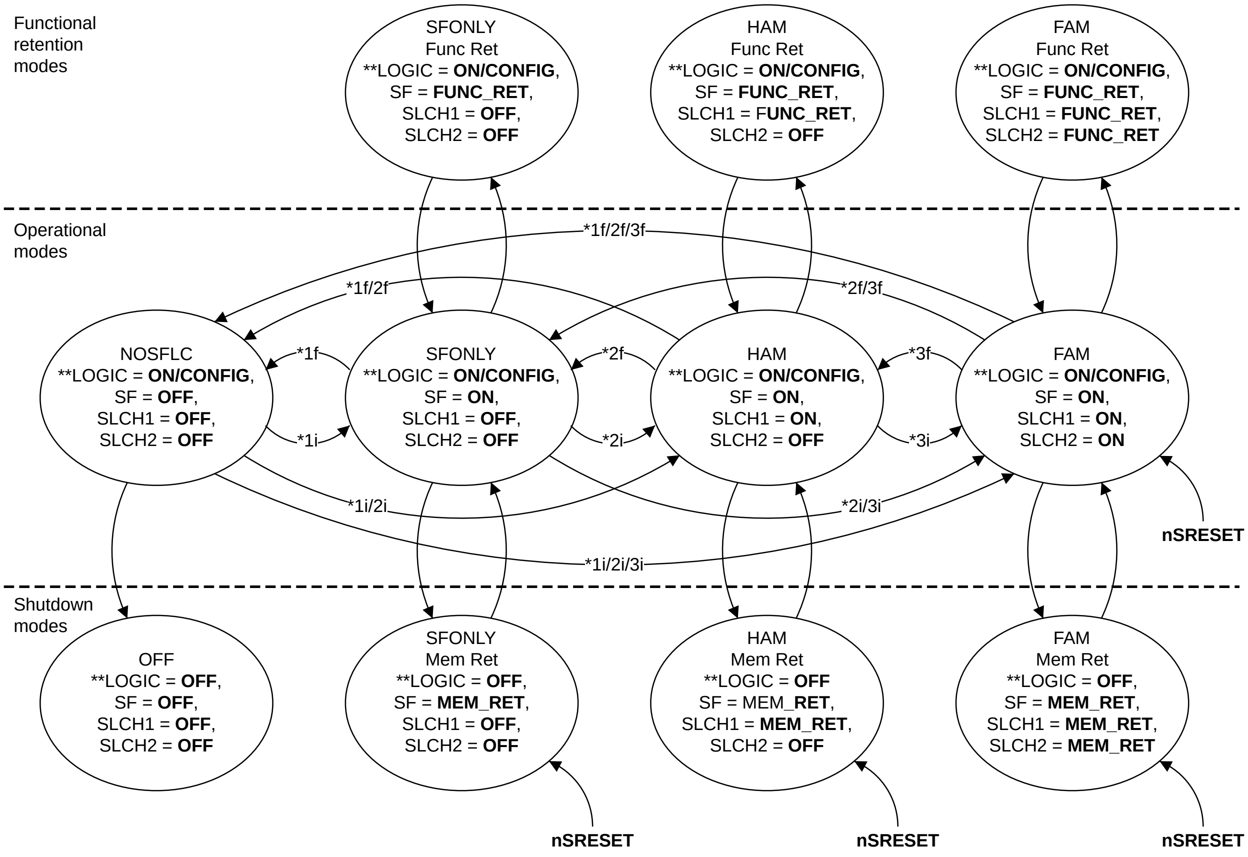
The following table shows the valid HN-F power states and their requirements.

Table 4-4: 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	Shut 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
OFF	Shutdown	Off	Off	Off	Off

The following figure shows the valid power states and transitions for a CMN-650 system.

Figure 4-12: Power state transitions


Functional retention modes
SFONLY Func Ret **LOGIC = ON/CONFIG, SF = FUNC_RET, SLCH1 = OFF, SLCH2 = OFF
HAM Func Ret **LOGIC = ON/CONFIG, SF = FUNC_RET, SLCH1 = FUNC_RET, SLCH2 = OFF
FAM Func Ret **LOGIC = ON/CONFIG, SF = FUNC_RET, SLCH1 = FUNC_RET, SLCH2 = FUNC_RET
Operational modes
*1f/2f/3f
*1f/2f
*2f/3f
NOSFLC **LOGIC = ON/CONFIG, SF = OFF, SLCH1 = OFF, SLCH2 = OFF
*1f
*1i
SFONLY **LOGIC = ON/CONFIG, SF = ON, SLCH1 = OFF, SLCH2 = OFF
*2f
*2i
HAM **LOGIC = ON/CONFIG, SF = ON, SLCH1 = ON, SLCH2 = OFF
*3f
*3i
FAM **LOGIC = ON/CONFIG, SF = ON, SLCH1 = ON, SLCH2 = ON
*1i/2i
*2i/3i
*1i/2i/3i
nSRESET
Shutdown modes
OFF **LOGIC = OFF, SF = OFF, SLCH1 = OFF, SLCH2 = OFF
SFONLY Mem Ret **LOGIC = OFF, SF = MEM_RET, SLCH1 = OFF, SLCH2 = OFF
HAM Mem Ret **LOGIC = OFF SF = MEM_RET, SLCH1 = MEM_RET, SLCH2 = OFF
FAM Mem Ret **LOGIC = OFF, SF = MEM_RET, SLCH1 = MEM_RET, SLCH2 = MEM_RET
nSRESET
nSRESET
nSRESET

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, 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.

To transition HN-F partitions to a required power state, write to the following por_hnf_ppu_pwpr register fields:

- policy
- op_mode

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

- pow_status
- op_mode_status

If either the SLC, the SF, or both are flushed as part of a power transition, then the power state transition can take many 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-650 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 in which 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-650 is fully off.

The HN-Fs automatically perform all activity that is required to enable safe transition between the respective power states in response to input P-Channel **PSTATE** transitions. It is not necessary for the SoC logic to perform any extra 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 as required by the respective power state transitions.

Note

CMN-650 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.

Note

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 4-5: 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

4.2.7 HN-F RAM Power Control State Machine interface

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

The following items depend on the completion of all P-Channel transactions:

- Each PCSM P-Channel interface that can be used to convert power state transitions into technology-specific controls
- The overall HN-F partition power state transition

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

Table 4-6: PSTATE Encodings

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

Note

This interface does not have a **PDENY** signal.

4.2.8 HN-F power domain completion interrupt

The PCCB can be configured to generate an **INTREQPPU** interrupt 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-650 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 the HN-F power transition completion sets all por_ppu_int_status register bits and the corresponding por_ppu_int_mask register bit.

To deassert **INTREQPPU**, write 1 to the bits of the por_ppu_int_status register that correspond to the masked group of HN-Fs that completed the power transitions.

4.2.9 RN entry to and exit from snoop and DVM domains

CMN-650 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-650.
- If an RN-F is excluded from the system coherency domain, it does not receive snoop or DVM requests from CMN-650.

RN-Ds behave as follows:

- If an RN-D is included in the system coherency domain, it must respond to DVM requests from CMN-650.
- If an RN-D is excluded from the system coherency domain, it does not receive DVM requests from CMN-650.

4.2.9.1 Hardware interface

This section describes the hardware interface for RN inclusion into and exclusion from system coherency domain.

CMN-650 provides two signals for RN system coherency entry and exit:

- **SYSCOREQ** input (to CMN-650)
- **SYSCOACK** output (from CMN-650)

These two signals implement a four-phase handshake between the RN and CMN-650 with four states. The following table describes this handshake.

Table 4-7: 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, an RN must assert **SYSCOREQ** and transition to the CONNECT state. The RN must be ready to receive and respond to snoop and DVM requests in this state.

ENABLED

Next, CMN-650 asserts **SYSCOACK** and transitions to the 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 the 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-650 deasserts **SYSCOACK** and transitions to the 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.

4.2.9.2 Software interface

This section describes the software interface for RN inclusion into and exclusion from system coherency domain.

CMN-650 provides two configuration registers for system coherency entry and exit:

RN-F

- por_mxp_p{1,0}_syscoreq_ctl
- por_mxp_p{1,0}_syscoack_status

RN-D

- por_rnd_syscoreq_ctl
- por_rnd_syscoack_status

Reading and writing to these configuration registers provides a software alternative to the four-phase hardware handshake.

Note

It is possible the configuration registers contain multiple bits where each bit corresponds to a different RN. The following description is about the read and write of the configuration register 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 configuration register bits corresponding to other RNs are not modified when writing into the syscoreq_ctl configuration register.

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

CONNECT

To initiate an RN entry into the system coherency domain, software must first poll both configuration registers. This poll ensures the configuration register 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 a 1 into the corresponding bit in the syscoreq_ctl configuration register. This write process transitions the RN to CONNECT state.

ENABLED

Next, CMN-650 indicates a transition to ENABLED state by setting the corresponding configuration register 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 configuration registers. After ensuring that the configuration register 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-650 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 the hardware interface or the 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 the hardware interface and enables the software interface. There must be a reset to re-enable hardware interface.
- When the software interface is employed, **SYSCOREQ** must remain deasserted.
- When the hardware interface is employed, software must not write to the syscoreq_ctl configuration register.

4.3 Identifying devices in the mesh

CMN-650 supports various types of IDs that are used to identify devices in the mesh. It uses these IDs, for example, for routing purposes.

4.3.1 Node ID mapping

The physical position of a device in the mesh determines the node ID that the device is mapped to.

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

Therefore, the device node ID is mapped to (X, Y, port, device ID).

Note

1. The size of the node ID field depends on the configured size of the mesh. Larger meshes have larger bit widths for the X and Y parameters.
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-650 mesh. The larger of the X and Y dimensions determines the size of the node ID. The following table shows the relationship between mesh X and Y dimensions and the node ID size.

Table 4-8: 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 contain the different node ID formats, depending on whether the node ID is 7 bits, 9 bits, or 11 bits wide.

Table 4-9: 7-bit node ID format

[6:5]	[4:3]	[2]	[1:0]
X position	Y position	Port	Device ID

Table 4-10: 9-bit node ID format

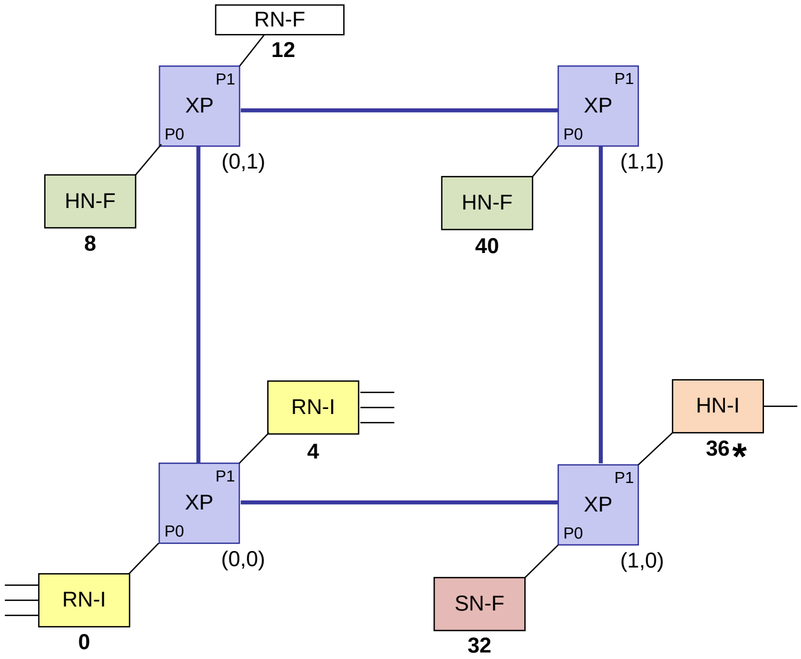
[8:6]	[5:3]	[2]	[1:0]
X position	Y position	Port	Device ID

Table 4-11: 11-bit node ID format

[10:7]	[6:3]	[2]	[1:0]
X position	Y position	Port	Device ID

The following figure shows a CMN-650 system with 7-bit node IDs in decimal format.

Figure 4-13: Example system with 7-bit node IDs



Example 4-1: 7-bit node ID format

For the HN-I connected to XP (1,0), the node ID reads as 36.

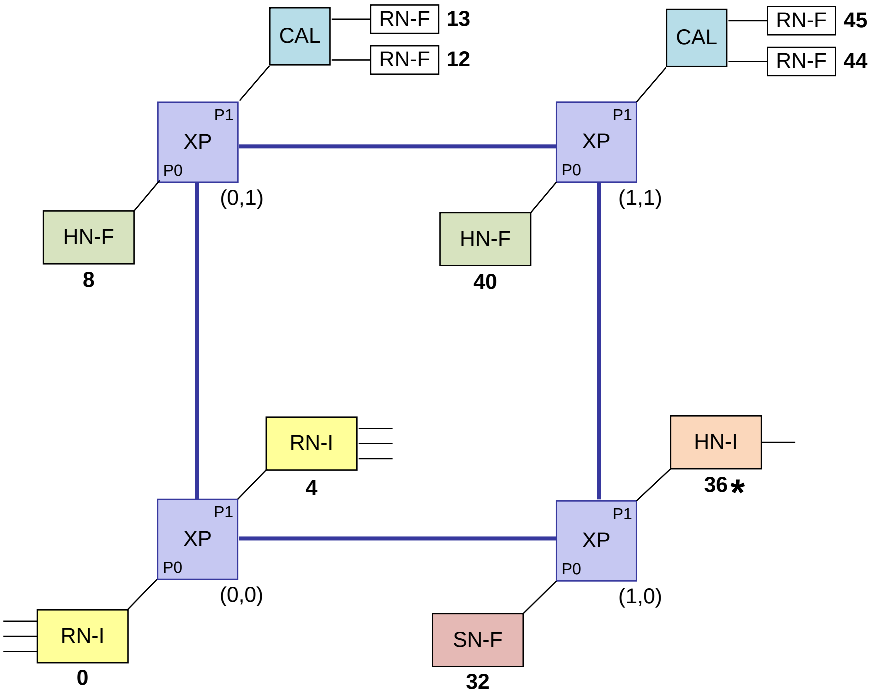
The equivalent binary value is `0b0100100`. In other words, the X position value = `0b01`, the Y position value = `0b00`, the device port value = `0b1`, and the device ID value = `0b00`.

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, bits [1:0] of the node ID are allocated according to the following scheme:

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

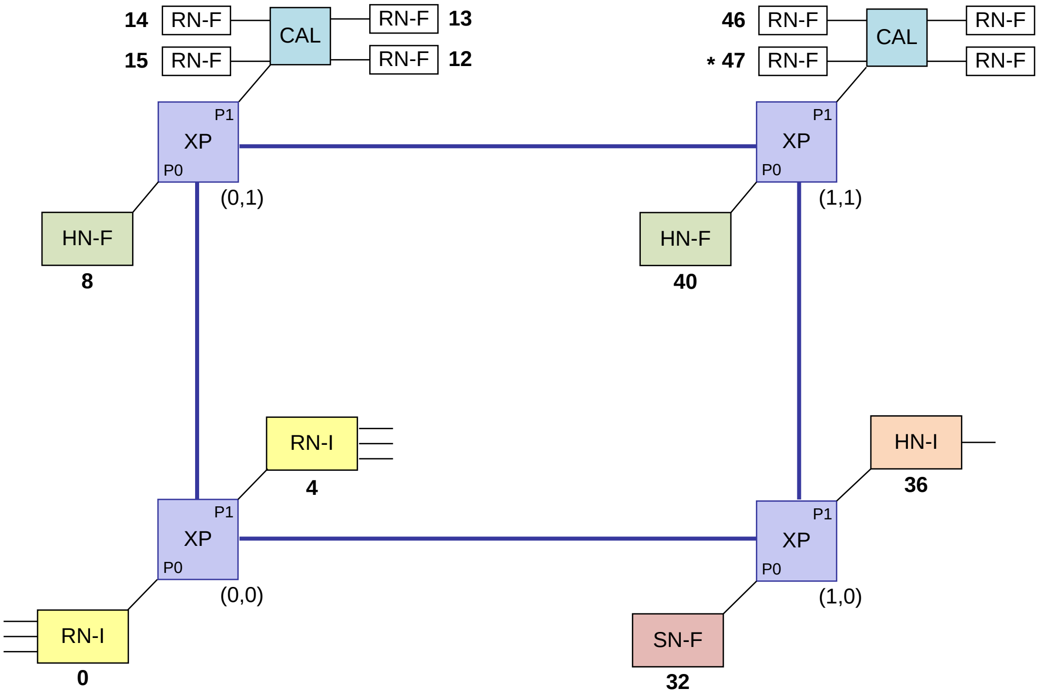
The following figure shows a CMN-650 system with CAL2 and 7-bit node IDs in decimal format.

Figure 4-14: Example system with 7-bit node IDs and CAL2



The following figure shows a CMN-650 system with CAL4 and 7-bit node IDs in decimal format.

Figure 4-15: Example system with 7-bit node IDs and CAL4



Example 4-2: 7-bit node ID format with CAL4

For the fourth RN-F connected to the CAL on XP (1,1), the node ID reads as 47.

The equivalent 7-bit node ID value is `0b0101111`. In other words, the X position value = `0b01`, the Y position value = `0b01`, the device port value = `0b1`, and the device ID value = `0b11`.

4.3.2 Logical Device IDs

In a CMN-650 interconnect, each instance of a device is assigned a *Logical Device ID* (LDID), which is used for various purposes. For example, the HN-F SF uses LDIDs to track cache lines.

LDIDs are unique within a particular device type. Therefore, two RN-Fs must not have the same LDID, although an RN-F and RN-I can have the same LDID. RN-Fs in a system could be assigned LDIDs 0-n, while RN-Is have LDIDs 0-m, and RN-Ds have LDIDs 0-k. The values n, m, and k depend on the number of each device type in the CMN-650 instance.

CMN-650 assigns default LDIDs at build time. If your configuration uses SF clustering, there are extra requirements related to LDIDs. For more information, see 4.11.4 LDID assignment when using SF clustering on page 156.

4.3.3 Extended CCIX Requesting Agent ID mechanism for up to 512 RN-Fs

CMN-650 CCIX RAs 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 6-bit base ID component of the RAID is based on the CCIX Agent ID, as defined in the *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*. The 4-bit expansion to the RAID enables CMN-650 to support up to 512 RAs in a system, while still being compatible with CCIX devices.

To be compliant with the CCIX specification, all uniquely identifiable agents on a CCIX device must be assigned unique IDs from the 6-bit base ID space. For CMN-650 RAIDs, the 6-bit base ID for all RAs on a chip must be unique to that chip. For example, consider a CML configuration where one CMN-650 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`.

Assigning RAIDs to devices in a mixed system

A mixed system can comprise the following types of devices:

- Devices that contain CMN-650 instances
- Devices that do not contain CMN-650, but that are compliant with revision 1.1 of the CCIX specification

Because the CMN-650 RAID is larger than the CCIX Agent ID, some assignment rules apply to these mixed systems. You must apply the following rules when assigning IDs to devices in a mixed system:

1. Assign a 6-bit unique CCIX Agent ID to each uniquely identifiable RA on all non-CMN-650 CCIX devices. For more information about assigning CCIX Agent IDs, including further rules and requirements that are not described in this document, see the *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*. These non-CMN-650 CCIX devices cannot use the 4-bit expanded ID space.
2. Identify all the CMN-650 agents that can communicate with the preceding CCIX agents and assign them a 6-bit unique ID. The expanded ID for these agents must be `0b0000` (`0x0`).
3. Assign all remaining CMN-650 agents an ID from the remaining 10-bit ID space.

Example 4-3: RAID assignment in an example mixed CML system

Consider a system configuration containing two CMN-650 chips, P0 and P1, and two accelerator chips, A0 and A1. The chips have the following characteristics:

P0 Chip is built using CMN-650, supports 10-bit RAID, and contains 128 RAs.

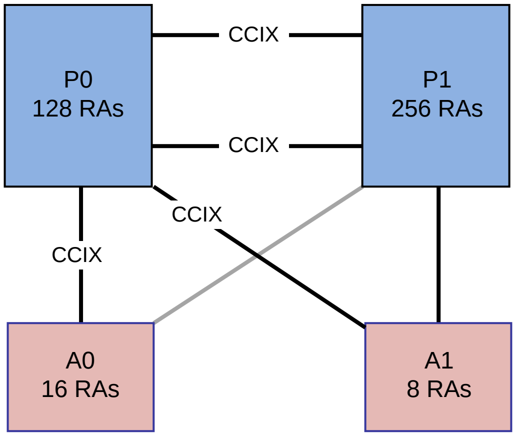
P1 Chip is built using CMN-650, supports 10-bit RAID, and contains 256 RAs.

A0 Chip is compliant with revision 1.1 of the CCIX specification, supports 6-bit CCIX Agent ID, and contains 16 RAs.

A1 Chip is compliant with revision 1.1 of the CCIX specification, supports 6-bit CCIX Agent ID, and contains 8 RAs.

The following figure shows this example CML system.

Figure 4-16: Example CML system



In this example system, RAIDs across P0, P1, A0, and A1, are assigned according to the following scheme:

1. A0 and A1 have 24 RAs in total. Therefore, they can be assigned RAIDs 0-23.
2. P0 has 128 RAs. P0 requires a minimum of eight unique base ID values because the 4-bit expanded ID space provides 16 unique RAIDs for each 6-bit base ID value. Therefore, you can find the number of unique base ID values that are required for a CMN-650 chip by dividing the number of RAs by 16, or 2^4.
3. P1 has 256 RAs. According to the calculation in the preceding step, P1 needs a minimum of 16 unique IDs from the base ID space.
4. We identify all the agents on P0 and P1 that can communicate with A0 and A1. The number of these agents can affect the final number of unique base IDs required for P0 and P1. In the example, 16 RAs from P0 and 8 RAs from P1 can communicate with A0 and A1.
5. Because P0 has 16 RAs that can communicate with A0 and A1, a total of 16 unique base IDs must be used for these RAs.
   - We assign IDs 24-39 to the RAs on P0 that can communicate with A0 and A1.
6. P1 has eight RAs that can communicate with A0 and A1, but the minimum number of unique base IDs that are required is 16, according to the number of RAs on the chip.
   - We assign IDs 40-55 to RAs on P1. Eight of these IDs must be assigned to RAs that can communicate with A0 and A1. The other eight can be assigned to any of the RAs on P1.
7. All remaining RAs which have not been assigned an ID on P0 and P1 can be assigned IDs from the remaining 10-bit ID space.

On P0 and P1, we must also allocate LDID values for all the RAs in the system. In the example system, we assign LDIDs in the CMN-650 device P0 in the following way:

1. 128 RAs on P0 are assigned LDIDs from 0-127.
2. 256 RAs from P1 are assigned LDIDs from 128-383.
3. 16 RAs from A0 are assigned LDIDs from 384-399.
4. 8 RAs from A1 are assigned LDIDs from 400-407.

We assign LDIDs in the CMN-650 device P1 in the following way:

1. 256 RAs on P1 are assigned LDIDs from 0-255.
2. 128 RAs from P0 can be assigned LDIDs from 256-383.
3. 8 RAs from A1 can be assigned LDIDs from 384-391.
4. 16 RAs from A0 can be assigned LDIDs from 392-407.

For more information about LDIDs in cross-chip routing, see the following sections:

- 4.11.2 Mapping LDIDs to RAIDs in CXRA and CXHA on page 153
- 4.11.3 Mapping SF LDID vector values to snoop targets in HN-F on page 155
- 4.11.4 LDID assignment when using SF clustering on page 156
- 4.11.5 Cross-chip routing examples on page 157

You can also override the default LDIDs when SF clustered mode is enabled. For more information about SF clustered mode and LDID override functionality, see the following sections:

- 4.11.4 LDID assignment when using SF clustering on page 156
- 6.2.11 Non-clustered and clustered mode for SF RN-F tracking on page 879
- 6.2.12 Configuring clustered mode for SF tracking on page 881

Although HAs are not shown in the example, all the chips, including A0 and A1, could have HAs. HAIDs can be same as any of the RAIDs assigned to the chip and the HAIDs must be from the unique base ID pool. In other words, the 4-bit expanded ID must be `0x0`.

CMN-650 RN-Is or RN-Ds that communicate on a non-SMP link must be given a unique 6-bit ID. This ID must not be the same as the RAID of any of the RAs on the chip.

CMN-650 RN-Is or RN-Ds that communicate on an SMP link can have the same RAID as any RA on the chip.

4.4 System Address Map (SAM)

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

The entire system address space can be partitioned into subregions. Each partition must be designated as one of the following types:

- I/O space
- DDR space

In CMN-650, HN-I, HN-D, HN-P, and HN-T nodes service requests to I/O space. HN-F, SN-F, and SBSX nodes service requests to DDR space.

The SAM maps regions in the system address space to the correct target. CMN-650 has software-configurable SAM blocks which allow a single implementation of CMN-650 to support programmable mappings of addresses to downstream targets.

Every master that is connected to CMN-650 must have the same view of memory. Therefore the SAM functionality is required for each requesting device, and each requesting device SAM must generate the same target ID value given the same address.

The SAM consists of several logical units:

RN SAM

Present for all RNs and CXHAs. Allows each RN to map addresses to HN-F, HN-I, HN-D, HN-T, and HN-P target IDs. The RN SAM also supports generation of SN-F and SBSX target IDs. The RN uses SN-F and SBSX target IDs to issue PrefetchTgt operations directly to the memory controller.

Note

Unmapped addresses are routed to the HN-D.

RA SAM

Present in all CXRA nodes. Maps addresses to CXHA node target IDs, in other words the HAID for the CXHA.

HN-F SAM

Present in all HN-Fs. Maps addresses to SN-F and SBSX target IDs.

HN-I SAM

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

Related information

- 4.5 RN SAM on page 96
- 4.6 RA SAM on page 115
- 4.7 HN-F SAM on page 116
- 4.9 HN-I SAM on page 129

4.5 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-650 RN-Fs, RN-Is, RN-Ds, and CXHAs use an RN SAM that is internal to the interconnect.

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.

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.

Related information

- 5.4.3 RN SAM and HN-F SAM programming on page 838

4.5.1 RN SAM memory regions and target types

When you set up the RN SAM, you configure it to map regions of the memory space to specific targets in the mesh. The RN SAM supports various memory region types and target types. It also has some predefined targets, including a default target, that are used in certain scenarios.

The CMN-650 RN SAM can contain the following memory region types:

- *Generic Interrupt Controller* (GIC) memory region
- Non-hashed memory region
- Hashed memory region

GIC memory region

The GIC memory region maps specific GIC-related addresses to a specific target ID. You can only specify one GIC memory region.

Non-hashed memory regions

Non-hashed memory regions are intended to target the I/O space of your system memory map. In other words, the HN targets of non-hashed regions are typically, but not always, one of the following HNs:

- HN-I
- HN-D
- HN-T
- HN-P

A non-hashed memory region always has a single target.

You can specify 8, 12, 16, or 20 non-hashed memory regions in the RN SAM non-hashed region registers.

Note

For RN-F ESAM node types, there is a latency consideration when using 12, 16, or 20 non-hashed regions in the RN SAM. Having 12, 16, or 20 non-hashed regions adds an extra cycle of latency for RN SAM lookup in the XP on the REQ flit.

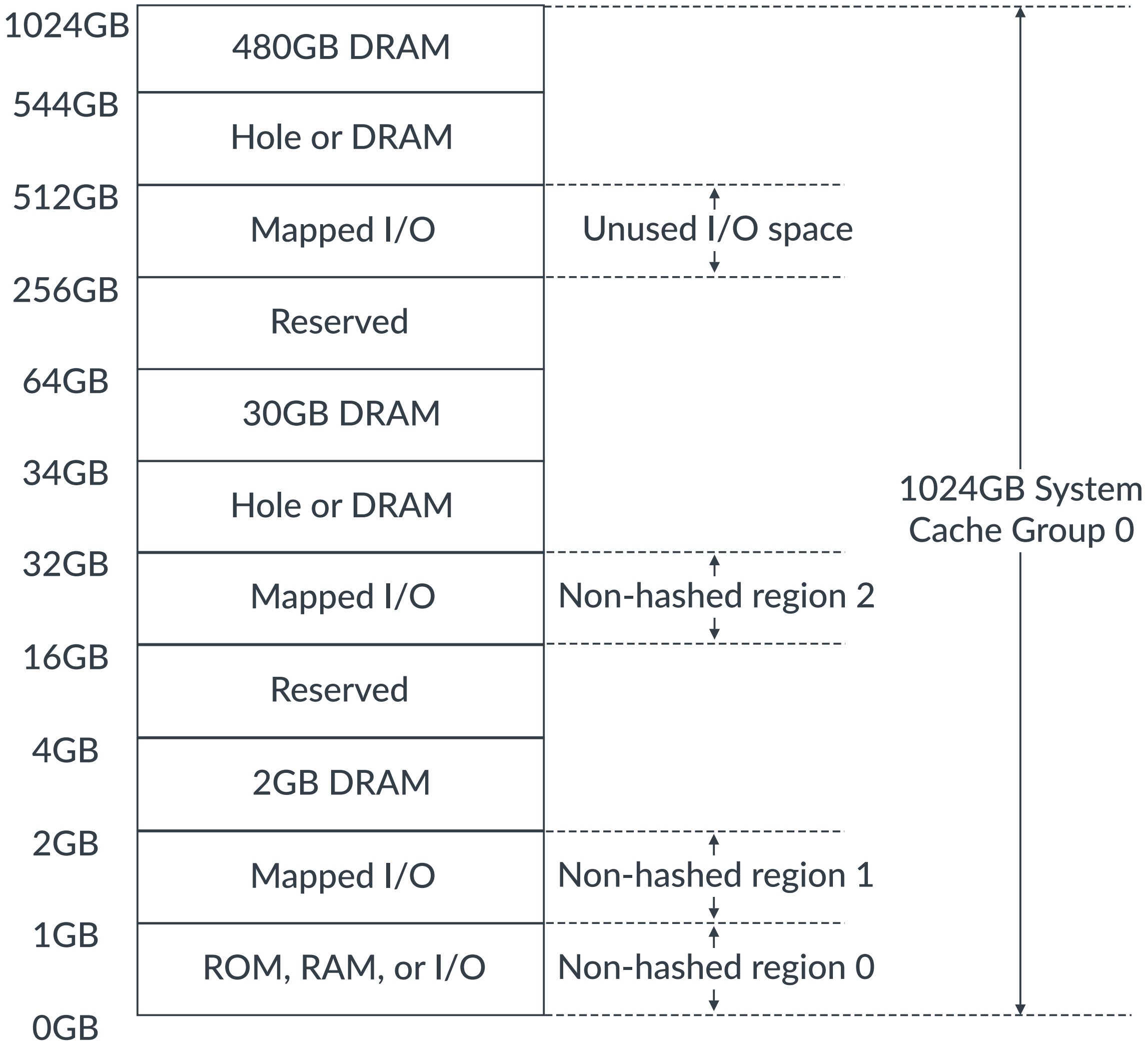
Hashed memory regions

Hashed memory regions are intended to target the DRAM space of your system memory map. In other words, the targets of hashed regions are typically, but not always, a group of HN-Fs. Accesses to hashed memory regions are distributed across multiple HN-Fs based on a passing the PA of the transaction through a hash function.

The group of HN-Fs that a hashed memory region targets are known as a *System Cache Group* (SCG). You can specify up to four SCGs in the RN SAM SCG region registers. Each SCG supports a primary and secondary address range, so the RN SAM supports up to eight hashed memory regions.

The following figure shows an example system memory map with the memory regions identified.

Figure 4-17: Example memory map



Memory regions of the same type cannot overlap with one another, but different memory regions of different types can overlap. For example, two non-hashed regions cannot overlap with one another, but a non-hashed memory region can overlap with a hashed memory region. A priority-based scheme determines the target of an access to an address that is in two overlapping regions. For more information, see 4.5.3 RN SAM target ID selection on page 103.

RN SAM memory region requirements

Each of the programmed region sizes must be a power of two and the region must be size-aligned. The region size can range from 64KB–4PB. For example, a 1GB partition must start at a 1GB-aligned boundary.

It is possible to support complex memory maps where DRAM region sizes are not a power of two or are not size-aligned. For example, Figure 4-17: Example memory map on page 99 shows a more complex memory map. The entire address space is assigned to a single hashed region, SCG0. Then, non-hashed regions are individually programmed within that space, which is possible because of their higher target ID selection priority. Software must prevent accesses to addresses in a hashed region that physical memory does not actually serve. For more information, see the *Principles of Arm® Memory Maps White Paper*.

Alternative configurations for non-hashed and hashed memory regions

Usually, non-hashed regions target the I/O space, and hashed regions target the DRAM space. However, alternative configurations are also possible. CMN-650 supports the following alternative memory region configurations:

- Using a hashed region to target only one HN-F, instead of multiple HN-Fs
- Using a non-hashed region to target an HN-F, and therefore DRAM. This scenario might be useful if all the SCG region registers have already been used. For target ID selection purposes, this HN-F is classified as a non-hashed target.
- Using an SCG region register for an HN-I, HN-D, HN-P, or HN-T target. This scenario might be useful if all the non-hashed region registers have already been used. Optionally, you can classify SCG regions 1, 2, or 3 as a non-hashed region. You cannot classify SCG region 0 as a non-hashed region.

Non-region-based targets

The RN SAM contains logic to map certain transaction types to the HN-D node regardless of the PA of the request.

During the boot process, before RN SAM programming, the RN SAM applies a default target ID to all transactions. This default target ensures that transactions can progress. All RN SAMs define the HN-D node as the default target. The interconnect sends all transactions to the HN-D until RN SAM programming completes, when the configured RN SAM regions take effect.

The RN SAM also assigns all DVM transactions to the HN-D, which contains the DN. Therefore, RN SAMs map the target ID of all DVM transactions to the nodeID of the HN-D.

Related information

- 4.5.2 SAM memory region size configuration on page 100
- 4.5.3 RN SAM target ID selection on page 103

4.5.2 SAM memory region size configuration

Hashed, non-hashed, and GIC 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:

Hashed and non-hashed

64MB for RN SAM and HN-F SAM up to maximum addressable space (2^{PA_WIDTH}).

GIC

64KB, 128KB, 256KB, and 512KB.

The following table shows the GIC memory region size encodings to program the RN SAM and HN-F SAM registers.

Table 4-12: RN SAM and HN-F SAM configuration register GIC memory region sizes

GIC memory region size	regionX_size value
64KB	3'b000
128KB	3'b001
256KB	3'b010
512KB	3'b011

The following table shows the hashed and non-hashed memory region size encodings to program the RN SAM and HN-F SAM registers.

Table 4-13: RN SAM and HN-F SAM configuration register hashed and non-hashed memory region sizes

Hashed and non-hashed memory region size	regionX_size value
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

Hashed and non-hashed memory region size	regionX_size value
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 target type encodings are listed in the following table.

Table 4-14: Device target types

Device type	Target type
HN-F	2’b00
HN-I	2’b01
CXRA	2’b10
Reserved	2’b11

The following table contains RA SAM configuration register memory partition sizes and encodings.

Table 4-15: 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

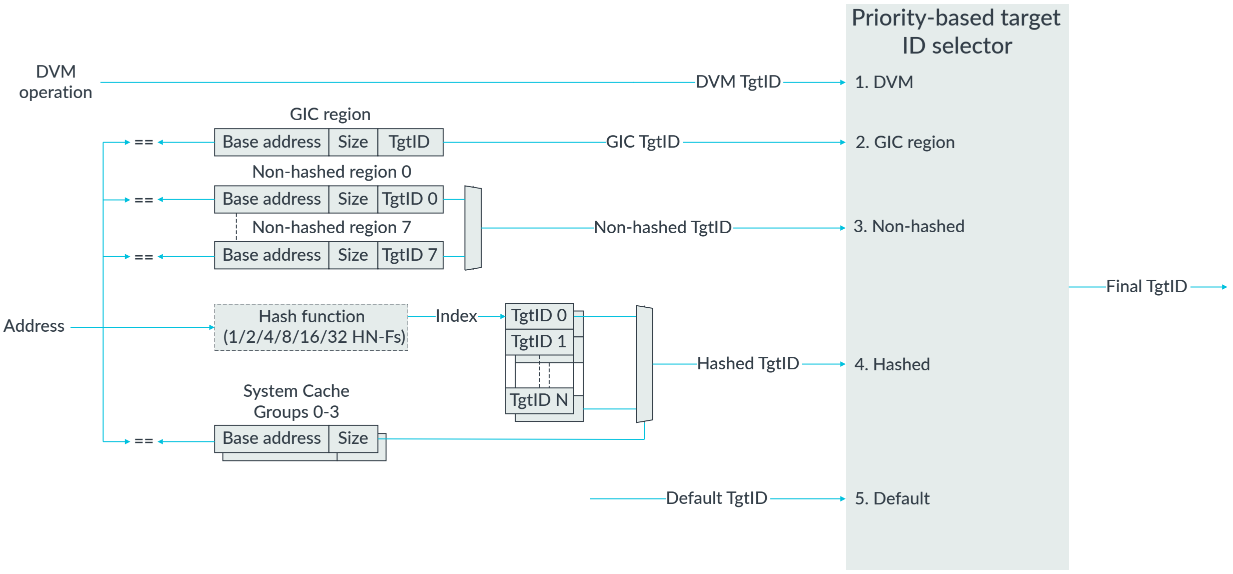
Memory partition size	regionX_size value
2TB	6'b011001
4TB	6'b011010
8TB	6'b011011
16TB	6'b011100
32TB	6'b011101
64TB	6'b011110
128TB	6'b011111
256TB	6'b100000
512TB	6’b100001
1PB	6’b100010
2PB	6’b100011
4PB	6’b100100

4.5.3 RN SAM target ID selection

The RN SAM uses a priority-based target ID selection scheme to select targets for specific addresses. This scheme determines the target ID for certain transaction types. It also resolves the target ID for a request with a PA that is within multiple overlapping memory regions in the RN SAM.

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 selects the target for the transaction based on a defined set of priorities. The following figure shows the RN SAM target ID selection policy.

Figure 4-18: RN SAM target ID selection policy



As the preceding figure shows, different targets have the following priority (from highest priority to lowest priority):

1. DVM target ID
2. GIC target ID
3. Non-hashed target ID
4. Hashed target ID
5. Default target ID

The DVM target has the highest priority. Therefore, if an RN sends a DVM request, the RN SAM always sends it to the DVM target ID. The DVM target ID is always the HN-D node.

The GIC target has the second highest priority. Therefore, the RN SAM always sends non-DVM requests that are in this region to the programmed GIC region target ID. The GIC memory region is higher priority than the non-hashed or hashed memory regions. Therefore, the RN SAM uses the GIC region target even if the GIC memory region overlaps with a non-hashed or hashed memory region.

The non-hashed targets have the third highest priority. Consider a non-DVM request with a PA that falls within the programmed non-hashed memory region. If the PA does not also fall into an overlapping GIC region, then the RN SAM sends the request to the non-hashed target. The RN SAM uses the non-hashed target even if the PA also falls into an overlapping hashed region, because of the higher non-hashed priority.

The hashed targets have the fourth highest priority. Consider a non-DVM request with a PA that falls within the programmed address range for an SCG. Usually, if the PA does not fall into an overlapping region with higher priority, the RN SAM hashes the PA to determine the target HN-F. Alternatively, if the hashed memory region is in non-hashed mode, then the RN SAM sends the request to the programmed target HN-F. Hashed mode and non-hashed mode for a given hashed region are mutually exclusive.

The default target ID has the lowest priority. Consider a non-DVM request with a PA that does not fall within one of the programmed address regions. In this case, the RN SAM sends the request to the default target ID. The default target ID is always the HN-D node. Read and write requests that do not target the CMN-650 configuration register space are issued on the HN-D ACE-Lite interface.

4.5.4 System Cache Groups (SCGs)

An SCG is a group of HN-Fs that share a contiguous memory region. A single HN-F in the SCG services a subset of addresses in the address region. To select the target HN-F in the SCG, the RN SAM uses a hash function.

Each HN-F in an SCG must have the same SLC size.

Each SCG supports two memory regions in the RN SAM, a primary and a secondary memory region. If an incoming address matches either of the two programmed, valid regions, then the RN

SAM selects an HN-F as the target ID for the request. To select the HN-F, the RN SAM hashes the PA bits of the request. The hash function outputs an index value that corresponds to a target HN-F ID in a table of target IDs.

The following limitations apply to the secondary memory regions for an SCG:

- Secondary memory region sizes must be size-aligned and power of two-sized.
- 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.

Related information

- 4.5.1 RN SAM memory regions and target types on page 97
- 4.5.5 SCG HN-F hash algorithm on page 105
- 4.5.6 Configuring SCGs in the RN SAM on page 106
- 4.5.7 Support for HN-Fs with CAL in the RN SAM on page 110

4.5.5 SCG HN-F hash algorithm

During RN SAM lookup, if the address falls within a hashed region, the SCG HN-F hash algorithm determines which HN-F services that request. The algorithm distributes addresses evenly across all HN-Fs in an SCG.

An SCG supports hashing over 1, 2, 4, 8, 16, 32, or 64 HN-Fs, using bits[MSB:6] of the PA of the request. If CMN-650 is configured to implement fewer than 52 PA bits, the unused upper bits are assumed to be zero.

In an SCG, each HN-F is given an index value. The size of the index depends on the number of HN-Fs in the SCG. For example, for an SCG that contains eight HN-Fs, the index is 3 bits wide. The hash algorithm distributes cache lines for the hashed address region between the different indexes, and therefore across all HN-Fs in the SCG.

The algorithm uses bits[MSB:6] of the PA of the request, and calculates the index value of the HN-F that services that address. The index value is associated with the node ID of the HN-F, so the index indicates which HN-F to send the request to.

The following table explicitly shows the hash algorithm.

Note

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.

Table 4-16: SCG HN-F hash function

Number of HN-Fs in SCG	Number of bits in index	Equations to calculate each index bit value
2	1	[0] = (6^7^8^...^51)

Number of HN-Fs in SCG	Number of bits in index	Equations to calculate each index bit value
4	2	[0] = (6^8^10^...^50) [1] = (7^9^11^...^51)
8	3	[0] = (6^9^12^...^51) [1] = (7^10^13^...^49) [2] = (8^11^14^...^50)
16	4	[0] = (6^10^14^...^50) [1] = (7^11^15^...^51) [2] = (8^12^16^...^48) [3] = (9^13^17^...^49)
32	5	[0] = (6^11^16^...^51) [1] = (7^12^17^...^47) [2] = (8^13^18^...^48) [3] = (9^14^19^...^49) [4] = (10^15^20^...^50)
64	6	[0] = (6^12^18^...^48) [1] = (7^13^19^...^49) [2] = (8^14^20^...^50) [3] = (9^15^21^...^51) [4] = (10^16^22^...^46) [5] = (11^17^23^...^47)

Related information

- 4.5.7 Support for HN-Fs with CAL in the RN SAM on page 110

4.5.6 Configuring SCGs in the RN SAM

The CMN-650 RN SAM supports up to four SCGs. The SCGs that can be used depend on the number of HN-Fs that you are allocating to each SCG.

The RN SAM supports up to 64 hashed HN-F and SN-F target IDs without using CAL mode. This feature allows up to 64 unique hashed target IDs in the RN SAM SCG target nodeID registers. RN SAM also supports up to 32 hashed target IDs when using CAL mode, with each target ID representing both HN-F nodes behind a CAL instance.

Restrictions on SCG selection

The number of SCGs that are available depends on the assignment of HN-Fs to certain SCGs. For example, the following table shows an example configuration of SCGs with 64 hashed target IDs. If

SCG0 uses 64 hashed HN-Fs, then SCGs 1, 2, and 3 cannot be used. Similarly, if SCG0 and SCG2 contain 32 HN-Fs each, then SCG1 and SCG3 cannot be used.

Table 4-17: Permitted allocation of HN-Fs into SCGs

SCG	Number of HN-Fs (or SN-Fs) per SCG						
	1 HN-F	2 HN-F	4 HN-F	8 HN-F	16 HN-F	32 HN-F	64 HN-F
SCG0	Y	Y	Y	Y	Y	Y	Y
SCG2	Y	Y	Y	Y	Y	Y	N
SCG1	Y	Y	Y	Y	Y	N	N
SCG3	Y	Y	Y	Y	Y	N	N

To set the number of HN-Fs in an SCG, program the por_rnsam_sys_cache_group_hn_count register. To assign HN-Fs (and SN-Fs for PrefetchTgt operations) to an SCG, program the sys_cache_grp_[hn, sn]_nodeid_reg<X> registers. Each sys_cache_grp_[hn, sn]_nodeid_reg<X> register contains space for four nodeIDs.

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 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.

Example SCG configurations

For example, the following table shows the possible register and nodeID allocation for each SCG in a system with 64 hashed target IDs.

Note

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

Table 4-18: RN SAM SCG target ID programming for 64 hashed targets

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					-

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_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
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 possible register and nodeID allocation for each SCG in a system with 16 hashed target IDs.

Table 4-19: RN SAM SCG target ID programming for 16 hashed targets

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 possible register and nodeID allocation for each SCG in a system with eight hashed target IDs.

Table 4-20: RN SAM SCG target ID programming for eight hashed targets

SCG target ID registers (eight 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
				-

SCG target ID registers (eight hashed targets)	Number of HN-Fs per SCG target ID table			
	8	4	2	1
			SCG3_NIDs	SCG3_NIDs
				-

The following table shows the possible register and nodeID allocation for each SCG in a system with four hashed target IDs.

Table 4-21: RN SAM SCG target ID programming for four hashed targets

SCG target ID registers (four 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 following table contains an example mapping of HN-Fs to SCGs in a configuration with 25 HN-Fs.

Table 4-22: 25 HN-Fs to three SCGs programming example

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

The following table contains example programming for 25 HN-Fs.

Table 4-23: Example programming for 25 HN-Fs

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				-

Non-hashed mode for SCGs

SCGs 1-3 can be configured to non-hashed mode. In non-hashed mode, the SCG can contain a single HN-I, HN-T, HN-D, or HN-F. A separate register is used to define the target HN node ID.

Related information

- 4.5.7 Support for HN-Fs with CAL in the RN SAM on page 110
- 5.3.7.11 sys_cache_group_hn_count on page 450
- 5.3.7.14 sys_cache_grp_hn_nodeid_reg0-15 on page 454
- 5.3.7.13 sys_cache_grp_nonhash_nodeid on page 453
- 5.3.7.15 sys_cache_grp_sn_nodeid_reg0-15 on page 455

4.5.7 Support for HN-Fs with CAL in the RN SAM

In CMN-650, you can pair two HN-Fs at an MXP device port using a CAL. If you have HN-Fs with CAL in your configuration, there are extra configuration decisions to make when setting up the RN SAM.

CMN-650 supports up to 32 CAL instances for HN-F nodes in a mesh, allowing up to 64 HN-F nodes.

The device ID (NodeID[1:0]) field is the only thing that differentiates the NodeIDs for the HN-Fs that are paired at a CAL. For more information, see 4.3.1 Node ID mapping on page 88.

Therefore, there are several options for assigning HN-F nodes to SCGs when CALs are used:

Note

The following description of these options uses the example configuration:

- Four CAL instances are used to connect eight HN-F nodes (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 4.5.6 Configuring SCGs in the RN SAM on page 106 describes. In the example configuration, all eight HN-F node IDs would be entered in the target ID registers of the SCG. Therefore, each HN-F has an entry in the hashed target ID table.
The HN-F count field for that SCG would be set to eight. This approach allows up to 64 HN-F nodes that are connected to 32 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 would be entered in the SCG target ID registers (one per CAL). The HN-F count register for that SCG would be set to four. This approach allows up to 64 HN-F nodes that are connected to 32 CAL instances to be assigned to an SCG.

Mixed mode

In systems with mixed CAL and non-CAL HN-Fs, you can assign a mix of CAL and non-CAL HN-Fs to a single SCG. The CAL HN-Fs must be individually programmed in normal mode.

You must not enable the scg<X>_hnf_cal_mode_en field in the sys_cache_grp_cal_mode_reg register for this kind of SCG.

Note

Extra configuration considerations might apply in the following scenario:

- All HN-F nodes are attached to CAL instances in your configuration.
- You intend to use normal mode to assign the target node IDs to SCGs.

Specifically, it might be necessary to modify the *RNSAM_NUM_ADD_HASHED_TGT* global RTL parameter when you configure the mesh in Socrates™. By default, the number of HN-F CAL instances determines the number of sys_cache_grp_hn_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. You must increase the value by the number of HN-F CAL instances that are present in the mesh.

The RN SAM sys_cache_grp_cal_mode_reg register contains the CAL mode enable bit for each SCG. For example, to enable CAL mode for SCG 0, write 1 to bit[0] of this register.

The NodeID[1:0] field differentiates the HN-Fs that are attached to a CAL. In CAL mode, the RN SAM uses this differentiation to hash addresses over twice the number of HN-Fs using the existing hashed target ID registers.

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 index = 3

[0] = (6^9^12^15^18^21^24^27^30^33^36^39^42^45^48^51)
[1] = (7^10^13^16^19^22^25^28^31^34^37^40^43^46^49)
[2] = (8^11^14^17^20^23^26^29^32^35^38^41^44^47^50)

Index bits[1:0] are used to pick between the programmed four HN-F target IDs. Index bit[2] is used to override the device ID field in the following manner:

Target NodeID[10:0] = {hash_nodeID_pick[10:1], hash_nodeID_pick[0] ^ index bit[2]}.

Therefore, in CAL mode, index bit[2] selects between the two HN-Fs on the CAL, which have consecutive NodeIDs.

The following limitations apply to the CAL mode in the RN SAM:

- This feature must only be used when the target HN-Fs are paired using CAL.
- RN SAM does not apply this method to SN-F target IDs. To fully utilize CHI-B PrefetchTgt operations to SN-F, you must map HN-Fs in CAL mode to SN-Fs in a specific way. 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 one 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 CXG NodeIDs, HN-F CAL mode must not be used for that SCG.

4.5.8 Address bit masking in the RN SAM

The CMN-650 RN SAM supports masking of the address bits that are used for address range comparison and address hashing.

To enable address bit masking in the RN SAM, program the following registers:

- rnsam_region_cmp_addr_mask_reg
- rnsam_hash_addr_mask_reg

When the RN SAM compares the incoming address against the programmed address 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].
- For the GIC memory region, RN SAM uses address bits[MSB:16].

By programming select bits to 0 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.

For hashed regions, the RN SAM hashes all address bits[MSB:6] to equally distribute requests across all HN-F and SN-F target devices. By programming select bits in the rnsam_hash_addr_mask_reg mask register to 0, those address bits can be removed from the hashing logic. This masking only applies to hashed memory regions.

The following limitations apply to address bit masking in the RN SAM:

- If 3-SN, 5-SN, or 6-SN mode is enabled in the HN-F SAM, address bits[16:8] and the top_addr_bits are essential in distributing the addresses between memory. We recommend that you mask these bits to avoid memory aliasing.
- The 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] must not be masked. Similarly, if a region size is 512MB, address bit[29] must not be masked.
- The address bits that are masked in the HN-F SAM and the RN SAM must be consistent. This requirement ensures that the same SN-F processes both PrefetchTgt requests to the address from an RN-F and SLC misses for the address from an HN-F.

Related information

- 4.5.3 RN SAM target ID selection on page 103
- 5.3.7.9 rnsam_hash_addr_mask_reg on page 448
- 5.3.7.10 rnsam_region_cmp_addr_mask_reg on page 449

4.5.9 Support for PrefetchTgt operations in RN SAM

The RN SAM supports CHI PrefetchTgt operations. These operations are sent from RN-F directly to SN-F, bypassing the HN-F.

To support PrefetchTgt operations, the RN SAM integrates the functionality of HN-F SAM to determine the appropriate SN-F target ID for a given address. RN-Fs only use the SN-F target ID for PrefetchTgt requests. The PrefetchTgt RN SAM programming must match the HN-F SAM for SN-F target IDs.

The register resources that are used to configure PrefetchTgt operations depend on how the HN-F SAM is configured for the associated address range. One set of registers is used for SN-F targets that are direct-mapped or address-striped in the HN-F SAM. The other is used for address range-based SN-F targets.

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

The following registers are used to program the direct-mapped or address-striped PrefetchTgt 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}

The PrefetchTgt RN SAM registers are only present in RN SAM blocks that are associated with RN-F nodes.

Note

For RN SAM blocks associated with other node types such as RN-I, RN-D, and CXHA:

- Reads of these register offsets always return a value of zero.
- Writes to these register offsets have no effect.
- These registers do not appear in the IP-XACT files.

PrefetchTgt operations to address range-based SN-F targets

The sam_qos_mem_region_reg* registers can be used to send PrefetchTgt operations to range-based targets in the HN-F SAM. These registers are also used to override the QoS value of requests from RN to HN. For information about enabling PrefetchTgt operations to range-based targets and the relationship with the QoS override feature, see 4.5.10 Address range-based QoS override and PrefetchTgt support on page 113.

Related information

- 4.7 HN-F SAM on page 116

4.5.10 Address range-based QoS override and PrefetchTgt support

The CMN-650 RN SAM provides resources that can override the QoS value of requests and facilitate PrefetchTgt 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 of these features are enabled. The following table shows the possible modes and their respective values.

Table 4-24: QoS override and PrefetchTgt 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 PrefetchTgt mode enabled

The QoS override mode and the PrefetchTgt 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 CXHA.

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

The QoS regions follow the same base address and size properties as hashed and non-hashed regions.

When you enable the address range-based QoS override feature, CMN-650 compares the incoming address against the valid QoS override memory regions. If the address matches any of the programmed valid addresses, the REQ 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.

Support for PrefetchTgt operations to address range-based targets

Setting the *RNSAM_NUM_QOS_REGIONS* parameter to 8 enables the PrefetchTgt option. If this option is enabled, and the incoming address matches one of the specified QoS regions, the sn_tgtid_override bit for the region is set to 1. If the address of the request is in the corresponding SCG, the RN SAM generates the SN target ID as programmed in sys_cache_grp_region[0-1]_sn_nodeid_reg[0-15] registers.

The node IDs of the SNs in these registers are mapped to each SCG similarly to the node IDs of the HNs. 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 por_hnf_sam_memregion[0-1] registers.

Table 4-25: Mapping of SCGs to QoS region registers

SCG ID	Mapped registers
SCG0	• sam_qos_mem_region_reg0 • sam_qos_mem_region_reg1
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

Related information

- 2.4 Global configuration parameters on page 23
- 4.20.2.1 QoS regulators on page 214
- 5.3.7.27 sam_qos_mem_region_reg0-7 on page 476
- 5.3.7.28 sys_cache_grp_region0_sn_nodeid_reg0-15 on page 477
- 5.3.7.29 sys_cache_grp_region1_sn_nodeid_reg0-15 on page 479

4.6 RA SAM

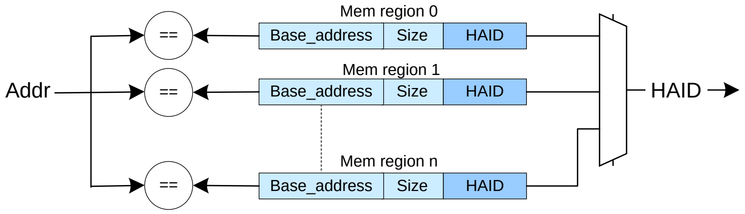
All *CCIX Requesting Agents* (CXRAs) in CMN-650 require a CCIX *Requesting Agent System Address Map* (RA SAM) to determine the target CCIX *Home Agent ID* (HAID). This HAID is used as the target ID to route the CCIX request.

The RA SAM uses configuration registers to specify address regions. The registers associate a specific address region with a corresponding target HAID. You configure the address regions by programming the base address and corresponding size of the address region into the por_cxg_ra_sam_addr_region_reg_0-7 registers, or programming the address limit. There is a valid bit to mark each valid address region is marked.

When the regions are marked as valid, the RA SAM compares incoming address against the programmed regions to generate a specific HAID.

The following figure shows the RA SAM target HAID generation logic.

Figure 4-19: RA SAM target HAID generation logic



Related information

- 5.3.2.8 por_cxg_ra_sam_addr_region_reg0-7 on page 272
- 4.11.1 Routing transactions across multi-chip systems on page 152

4.6.1 RA SAM address region requirements

When you set up the programmable RA SAM address regions, there are specific requirements for these regions that you must be aware of.

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. 0GB-1GB or 1GB-2GB are valid 1GB partitions, but 1GB partitions starting at 1.5GB or 2.5GB are invalid.

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. The CHI target ID is used to route the transaction to the correct SN.

The HN-F SAM target of a transaction is based on one of two characteristics:

- The PA of the transaction
- The number of SNs that are downstream of the SCG that the HN-F belongs to

Software can configure the HN-F SAM. To configure the HN-F SAM, you program the HN-F SAM registers, defining address range-based targets and targets for HN-Fs within an SCG.

Related information

- 5.4.3 RN SAM and HN-F SAM programming on page 838

4.7.1 Mapping SN targets in the HN-F SAM

The HN-F SAM has various methods of mapping transactions to an SN target ID. These methods are either based on the PA of the transaction, or a direct association between an HN-F and an SN.

The HN-F SAM uses three methods of mapping transactions at the HN-F to a downstream SN:

- Address range-based mapping
- Hashed modes
- Direct-mapped mode

Note

The hashed modes and the direct-mapped mode are mutually exclusive. In other words, an SCG can only use one of the hashed modes or direct-mapped mode. The mode that the SCG uses depends on the number of target SNs that the SCG has.

Address range-based mapping

For address range-based mapping, you can specify up to two address regions in each HN-F SAM. For each address region, you also specify a specific SN target ID, which is the target for transactions in that address range. This mode is useful when you must explicitly map a partition of memory from the global DRAM to an individual SN, for example, an on-chip SRAM.

Hashed modes

In the hashed modes, transactions are striped according to PA across either three, five, or six SNs. These modes are used when the SCG targets three, five, or six SNs, and are also known as 3-SN mode, 5-SN mode, and 6-SN mode respectively.

Addresses within the address range of the SCG are striped at a 256B granularity between the selected SNs. The stripe function uses address bits[16:8], and an extra two (3-SN) or three (5-SN, 6-SN) user-defined address bits.

Direct-mapped mode

In direct-mapped mode, transactions across an SCG target a single SN, or are distributed across 2, 4, 8, 16, or 32 SNs. In other words, this mode is used for a group of SNs with a size that is a power of 2. In direct-mapped mode, you program each HN-F to target a single nodeID, which is known as SN0. To distribute accesses from an SCG across multiple SNs, you distribute target nodeIDs evenly across the HN-Fs in the SCG.

For example, consider an SCG that contains eight HN-Fs and targets eight SNs. In this configuration, the SN0 field of each HN-F would be programmed with a different SN node ID. If an SCG with eight HN-Fs targets four SNs instead, every two HN-F nodes would have the same SN0 field value.

A priority scheme determines the final target node ID. For more information about this priority scheme, see 4.7.2 HN-F SAM target ID selection on page 118.

Related information

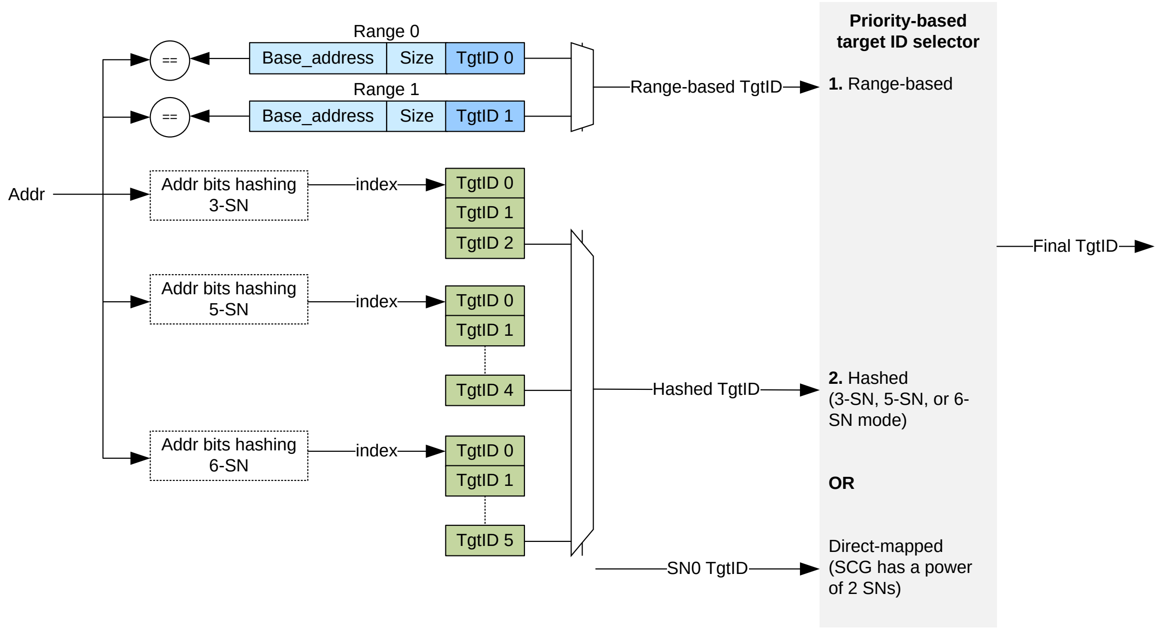
- 4.7.4 Example 3-SN, 5-SN, and 6-SN mode configurations on page 120

4.7.2 HN-F SAM target ID selection

A priority-based selection scheme determines the final target ID for any transaction that passes through the HN-F SAM.

The following figure shows how the target ID is determined from the HN-F SAM.

Figure 4-20: HN-F SAM target ID selection policy



As the preceding figure shows, different targets have the following priority (from highest priority to lowest priority):

1. Address range-based target
2. One of the following two mutually exclusive target types:
   - Hashed mode target
   - Direct-mapped mode target

The address range-based targets have the highest priority. Therefore, if the address falls into one of the two valid, programmed address ranges, the HN-F SAM always send the transaction to the address range-based target. Otherwise, one of the other two methods determines the target ID.

Whether the HN-F SAM uses one of the hashed modes or the direct-mapped mode depends on the number of downstream targets. The mode is defined across the whole SCG.

4.7.3 HN-F to SN-F memory striping in HN-F SAM

The CMN-650 HN-F SAM supports three 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.

3-SN mode

In 3-SN mode, a stripe function ensures that traffic is distributed evenly among the three SNs. The stripe function is based on PA[16:8] and two higher bits in the PA. The two higher PA bits are referred to as top_address_bit1 and top_address_bit0. Select the top address bits so that the following is true:

- Three of the four combinations of the top address bits appear evenly in the selected address space.
- 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:

SN = { ADDR[10:8] + ADDR[13:11] + ADDR[16:14] + ((top_addr_bit1<<1) | top_addr_bit0) } % 3

Example 4-4: Example SN distribution behavior

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 por_hnf_sam_control register to enable routing to three SNs. In the por_hnf_sam_control register, the hn_cfg_sam_top_address_bit0 and hn_cfg_sam_top_address_bit1 fields must be configured at boot time. These 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 SNs is selected using the following formula:

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 por_hnf_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 por_hnf_sam_control register along with hn_cfg_sam_top_address_bit1 and hn_cfg_sam_top_address_bit0 to hash the incoming address.

Related information

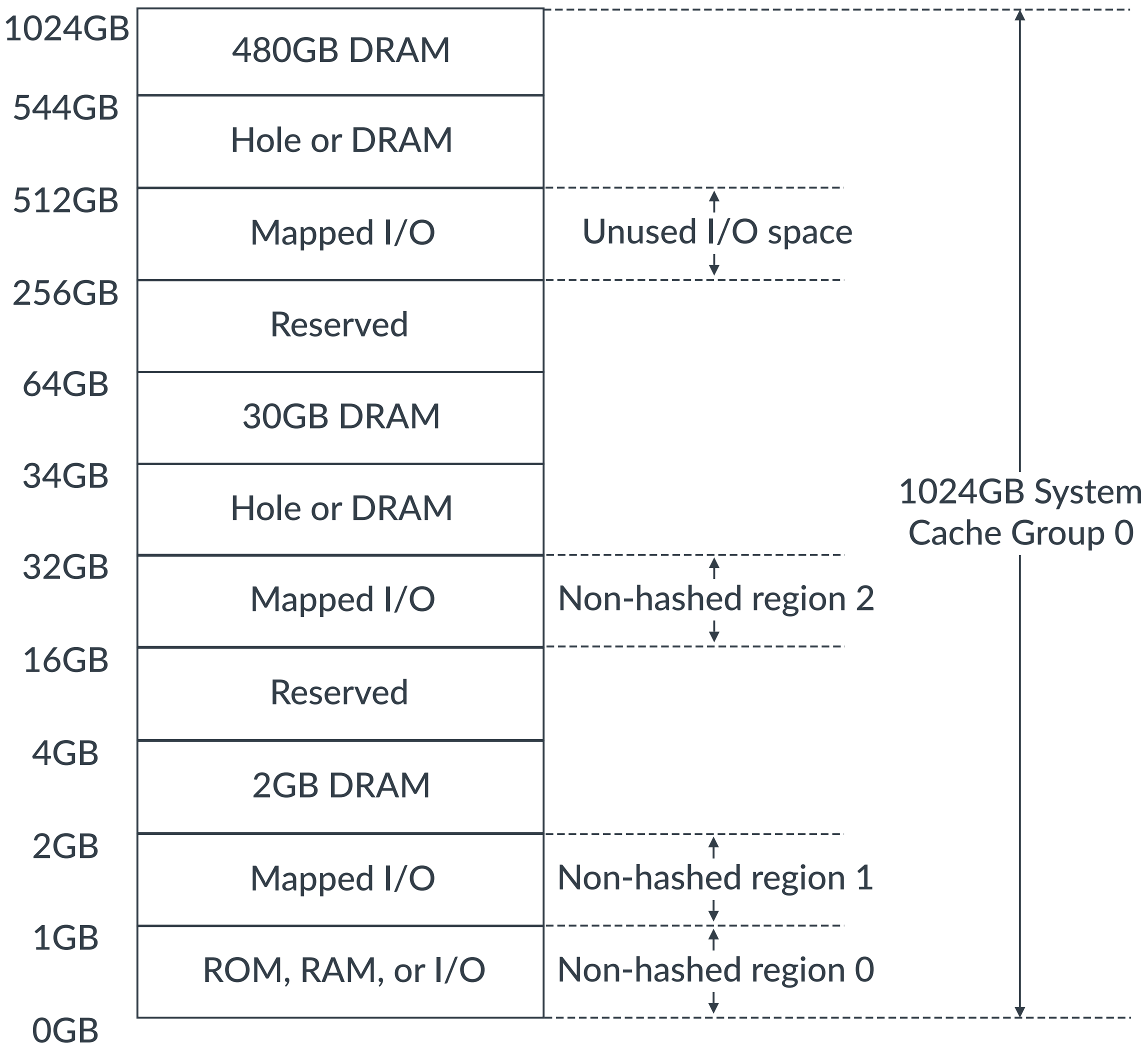
- 4.7.1 Mapping SN targets in the HN-F SAM on page 117
- 4.7.4 Example 3-SN, 5-SN, and 6-SN mode configurations on page 120

4.7.4 Example 3-SN, 5-SN, and 6-SN mode configurations

These hashed mode configuration examples use the Arm PDD memory map and show the configuration settings for each mode, including the top address bits.

The following figure shows the Arm proposed memory map.

Figure 4-21: Example memory map



The following table shows the valid top address bits for the Arm PDD memory map. For more information, see the *Principles of Arm® Memory Maps White Paper*. 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 as the following table shows.

Table 4-26: 3-SN mode top address bits [bit 1, bit 0]

Combination 1 (inv_top_address_bit set to 0)	Combination 2 (inv_top_address_bit set to 0)	Combination 3 (inv_top_address_bit set to 1)	Combination 4 (inv_top_address_bit set to 1)
[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 4-27: 6-SN mode top address bits [bit 2, bit 1, bit 0]

Combination 1 (inv_top_address_bit set to 0)	Combination 2 (inv_top_address_bit set to 0)	Combination 3 (inv_top_address_bit set to 1)
[0, 0, 0]	[0, 1, 0]	[0, 0, 0]
[0, 0, 1]	[0, 1, 1]	[0, 0, 1]
[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 4-5: 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. Since 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 first two regions together comprise a 32GB region, while the remaining two regions are 32GB each. The following table breaks down the address bits for the regions that are shown in the preceding list.

Table 4-28: Address region example bit settings: 3-SN example

Region	39	38	37	36	35	34	33	32	31
1	0	0	0	0	0	0	0	0	1
1	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 4-26: 3-SN mode top address bits [bit 1, bit 0] on page 122. 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 tables provide example address bits in 3-SN and 6-SN hashed modes with specific DRAM sizes. These address bits provide equal distribution of memory across all SN-Fs in the hashed group.

Table 4-29: 3-SN settings for specific DRAM sizes

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]	0
2GB (Total 6GB)	[35, 31]	0
4GB (Total 12GB)	[33, 32]	0
8GB (Total 24GB)	[34, 33]	0
16GB (Total 48GB)	[39, 34]	0
32GB (Total 96GB)	[39, 36]	1
64GB (Total 192GB)	[37, 36]	0
128GB (Total 384GB)	[38, 37]	0

Table 4-30: 6-SN settings for specific DRAM sizes

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]	0
2GB (Total 12GB)	[33, 32, 28]	0
4GB (Total 24GB)	[34, 33, 28]	0
8GB (Total 48GB)	[39, 34, 33]	0
16GB (Total 96GB)	[39, 36, 28]	1
32GB (Total 192GB)	[37, 36, 28]	0
64GB (Total 384GB)	[38, 37, 28]	0

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.

4.7.5 Maintaining contiguous address spaces in SN-Fs

To retain contiguous address spaces in each SN-F, SN-Fs typically remove one or more of the PA bits that are presented to them. The bits that the SN-F can strip from the PA depend on several factors that you specify in the RN SAM and HN-F SAM.

If all HN-Fs send their cache misses to a single SN-F, that SN-F services the full address space. However, it is common for a system to have two or more SN-Fs, with each HN-F sending cache misses to a single SN-F. In this scenario, each SN-F services only part of the address space. To keep a contiguous address map, SN-Fs typically remove one or more address bits. 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 SCG
- Number of SN-Fs in the SCG
- Which HN-Fs share SN-Fs

Direct-mapped SN targets

The following table shows direct-mapped HN-F and SN-F combinations that are supported within an SCG, along with the PA bits that we recommend removing.

Table 4-31: HN-F and SN-F combinations supported within an SCG

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]

Number of HN-Fs	Number of SN-Fs	Bits to strip from full PA
	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]

For direct-mapped HN-F and SN combinations, use the following method to calculate the bits to remove:

- Calculate the highest bit to remove. To find this bit, determine the least significant address bit in the most significant index bit of the HN-F hash function. The following bits are the highest bit to remove for all the possible SCG HN-F counts:

  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]

- The number of bits stripped is $\log_2$(number of SN-Fs). Remove bits sequentially below the highest bit that is calculated according to the preceding process.

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, consider an SCG configuration with eight HN-Fs targeting two SN-Fs. In this case, the bottom four HN-Fs in the RN SAM table would share one SN-F. Similarly, the top four HN-Fs would share the other SN-F.

3-SN, 5-SN, and 6-SN address striped targets

3-SN, 5-SN, and 6-SN address hashing modes implement a modulo function according to the top address bits used. Therefore, 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.

Related information

- 4.5.5 SCG HN-F hash algorithm on page 105

4.7.6 Address bit masking in the HN-F SAM

CMN-650 supports masking of address bits in the HN-F SAM. You can configure address bit masking for address region comparison and address hashing in hashed mode.

Address bit masking for address range-based targets in the HN-F SAM

The HN-F SAM uses PA bits[MSB:26] when comparing the incoming PA against the programmed address ranges in the HN-F SAM. You can mask off the incoming address and the programmed address ranges before comparison by programming select bits to 0 in the hn_sam_region_cmp_addr_mask_reg mask register.

This region mask is only applicable to address range-based memory partitioning in the HN-F. Therefore the mask is not applied to the hashing scheme in 3-SN, 5-SN, or 6-SN mode.

Address bit masking for hashed mode targets in the HN-F SAM

The HN-F SAM supports masking of the PA bits that are 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 mask register.

Limitations for PA bit masking in the HN-F SAM

The following limitations apply to PA bit masking in the HN-F SAM:

- The address range comparison 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, or 6-SN mode is enabled, address bits[16:7] and the top_addr_bits are essential to distributing the addresses between memory. We recommend that these bits are masked carefully to avoid memory aliasing.
- The address bits that are masked in the HN-F SAM and the RN SAM must be consistent. This requirement ensures that the same SN-F processes both PrefetchTgt requests to the address from an RN-F and SLC misses for the address from an HN-F.

Related information

- 4.7.1 Mapping SN targets in the HN-F SAM on page 117
- 5.3.15.62 hn_sam_region_cmp_addr_mask_reg on page 779
- 5.3.15.61 hn_sam_hash_addr_mask_reg on page 778

4.8 SAM support for CCIX Port Aggregation

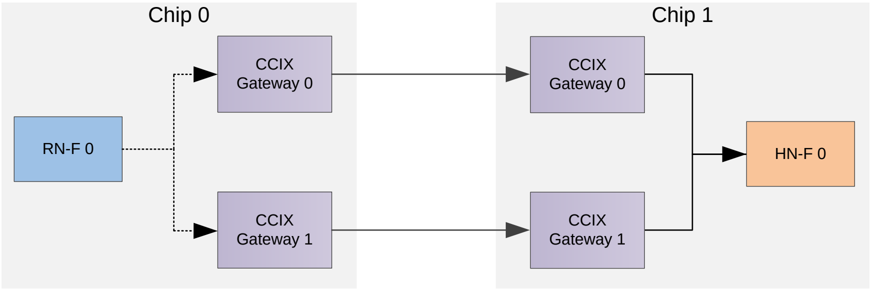
RN SAM and HN-F SAM both support *CCIX Port Aggregation* (CPA) functionality.

To determine whether CPA is enabled, RN SAM uses the address ranges and HN-F SAM uses the *Logical Device ID* (LDID) of the RN-F.

Support for CPA in RN SAM

Requests from an RN to a remote chip can be striped across CXGs according to PA as the *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0* describes. The following figure shows this functionality.

Figure 4-22: RN SAM CCIX Port Aggregation



This striping is achieved by hashing physical address bits[51:6]. The RN SAM can hash incoming addresses across up to ten gateway blocks forming a *CCIX Port Aggregation Group* (CPAG). RN SAM can support up to five CPAGs. 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 `0x3FFFFFFFFFF8`. With this mask setup, the logical operator AND is applied to all incoming addresses before hashing the address bits.

The following table shows the number of CXG targets that are allowed in each CPAG.

Table 4-32: Number of CXG targets allowed in each CPAG

CPAG ID	Number of CXGs supported
CPAG_0	8
CPAG_1	2
CPAG_2	4
CPAG_3	2
CPAG_4	2

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 of 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.

Table 4-33: Mapping of target IDs to CPAGs

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 CCIX gateway block.

HN-F uses the LDID of the RN-F to determine whether CPA is enabled. As 4.11.3 Mapping SF LDID vector values to snoop targets in HN-F on page 155 describes, HN-F contains an LDID to physical node ID conversion table. Table 4-43: Example LDID to target node ID programming on page 155 shows an example of this 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 the logical to physical ID conversion registers in HN-F.

Guidelines for enabling CPA in RN SAM and HN-F SAM

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

- CPA can only be used for SCG and non-hashed address ranges in the RN SAM.
- CPA must not be enabled for Device memory traffic, and can only be enabled for Normal memory traffic. For more information, see the *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*.
- 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 SCG 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.
- Each SCG in RN SAM can contain only one CPA group along with local HN-F target IDs.

- The cml_port_aggr_grp_reg0 and cml_port_aggr_grp_reg1 registers contain the full list of CXG target IDs.

Related information

- 5.5.3.11 Program CPA functionality in RN SAM on page 860
- 5.5.3.12 Program CPA functionality in HN-F SAM on page 861

4.9 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 that is connected to the HN-I can be one of the following types:

- Peripheral with memory-mapped I/O space, such as UART or GPIO
- Physical memory, such as SRAM or FLASH

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 because writes can be dispatched on AXI out of order.

OWO is still supported from RN-I, RN-D, and CXHA nodes.

To map and order the address space of these endpoints, the HN-I SAM supports several address regions. Each address region within the HN-I SAM can have its own specific properties, such as memory type.

The HN-I SAM has a default address region, which is known as address region 0. It can also contain up to three other address regions, which are known as address regions 1, 2, and 3.

Note

Address regions 1, 2, and 3 must not overlap.

Each address region in the HN-I SAM is also 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, 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.

Related information

- 4.9.1 HN-I SAM address region 0 on page 130
- 4.9.2 Configuring HN-I SAM address regions and order regions on page 130

4.9.1 HN-I SAM 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 one of the following options:

- 6'b101000 when *REQ_ADDR_WIDTH*==52
- 6'b100100 when *REQ_ADDR_WIDTH*==48
- 6'b100000 when *REQ_ADDR_WIDTH*==44

To configure the properties of address region 0, you program the por_hni_sam_addrregion0_cfg register. Address region 0 is always valid. Therefore, this register does not define a valid bit.

Related information

- 5.3.3.5 por_hni_sam_addrregion0_cfg on page 300

4.9.2 Configuring HN-I SAM address regions and order regions

To configure the HN-I SAM address regions and order regions, program the por_hni_sam_addrregion{0-3}_cfg registers. These registers contain several fields which control specific properties of the address regions and specify order regions. There are also some constraints on programming these registers.

Note

Arm recommends that the HN-I SAM is only programmed during the boot process.

Each address region can be programmed as either physical or peripheral memory. By default, each address region is mapped to peripheral memory.

The different types of memory have the following properties:

Physical memory

Follows normal memory ordering guarantees. Therefore, order region programming does not affect ordering for a physical memory address region.

Peripheral memory

- 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.

An address region register must be disabled if the following conditions are true:

- There is potential for new requests to fall into the address region register that is newly configured
- These new requests require ordering in relation to the existing outstanding requests

The minimum address granularity for address regions and order regions is 4KB. This size is equivalent to the minimum slave address space granularity in AXI and ACE-Lite. Therefore, the base address in the por_hni_sam_addrregion{1-3}_cfg registers only includes bits[*REQ_ADDR_WIDTH*-1:12].

Related information

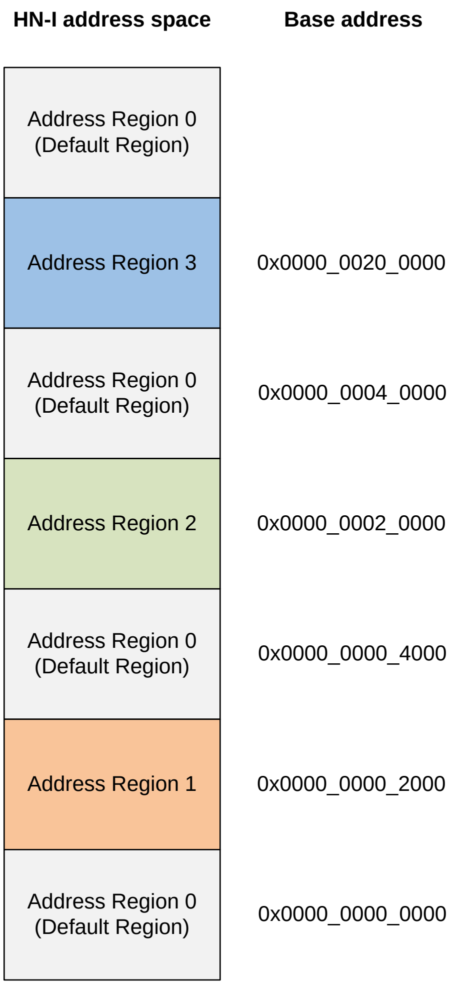
- 5.3.3.5 por_hni_sam_addrregion0_cfg on page 300
- 5.3.3.6 por_hni_sam_addrregion1_cfg on page 301
- 5.3.3.7 por_hni_sam_addrregion2_cfg on page 302
- 5.3.3.8 por_hni_sam_addrregion3_cfg on page 304

4.9.3 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 4-23: HN-I address space example



Note

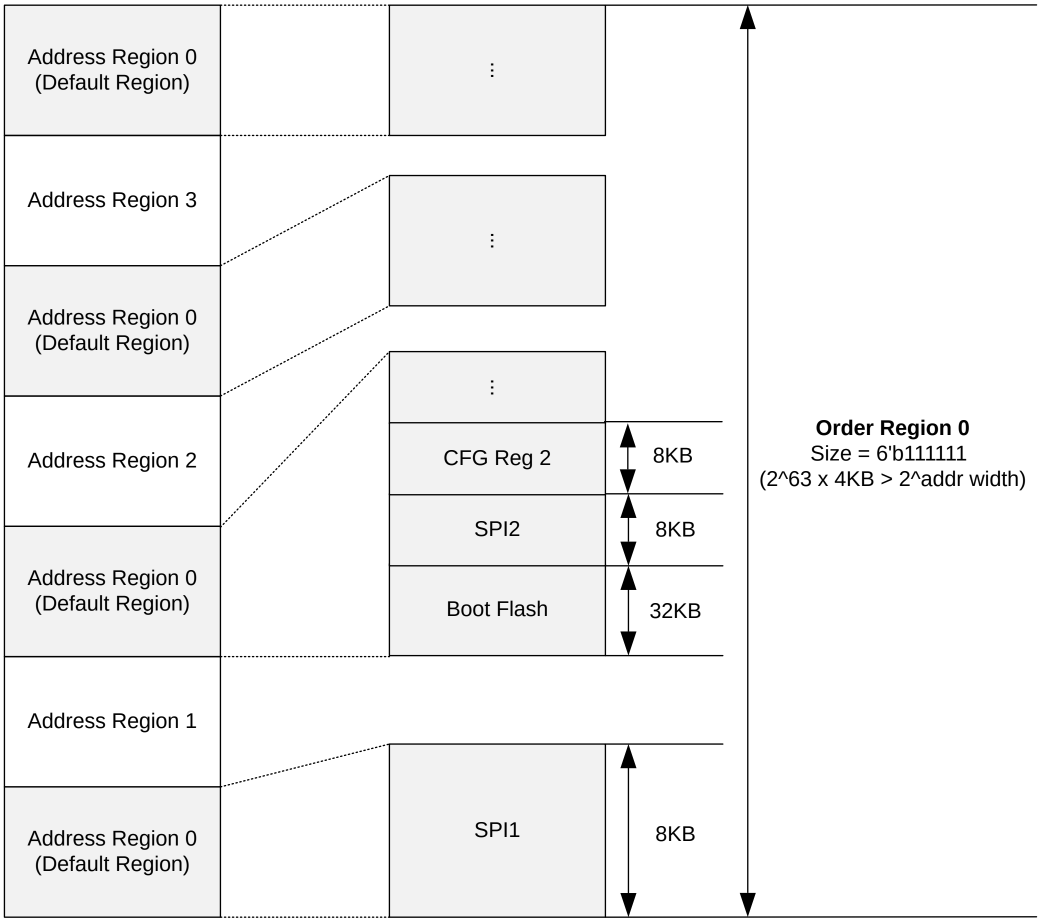
For this example, in the por_hni_sam_addrregion{0-3}_cfg registers, the following bit fields use the default value:

- ser_all_wr
- ser_devne_wr
- pos_early_wr_comp_en
- pos_early_rdack_en

Address region 0

In this example, address region 0 contains two SPI regions, SPI1 and SPI2, a boot flash region, and a configuration register region. The following figure shows this example address region 0 configuration.

Figure 4-24: Example address region 0 configuration



The largest peripheral address region in address region 0 is the boot flash region, which is 32KB. Requests targeting this boot flash region must be kept in order. If we configure the order region 0 size to 32KB, then requests with addresses from 0x0000_0000_0000 to 0x0000_0000_8000 are ordered. However, the boot flash is not aligned to a 32KB boundary, and spans 0x0000_0000_4000 to 0x0000_0000_C000). Therefore, part of the boot flash region sits between two order regions. Requests targeting that region might be kept out of order and cause issues. To ensure that the whole boot flash region is covered by a single order region, we must configure the order region 0 size at least 64KB.

However, if the order region 0 size is 64KB, then the SPI1, SPI2, and configuration register regions also fall into the first order region. Therefore, the following alternative configuration might be useful to optimize performance:

- Address region 0 has a memory map where boot flash is aligned to a 32KB boundary.
- The order region 0 size is configured to 32KB.

The following table shows the configured values for the por_hni_sam_addrregion0_cfg register when the order region size is 64KB.

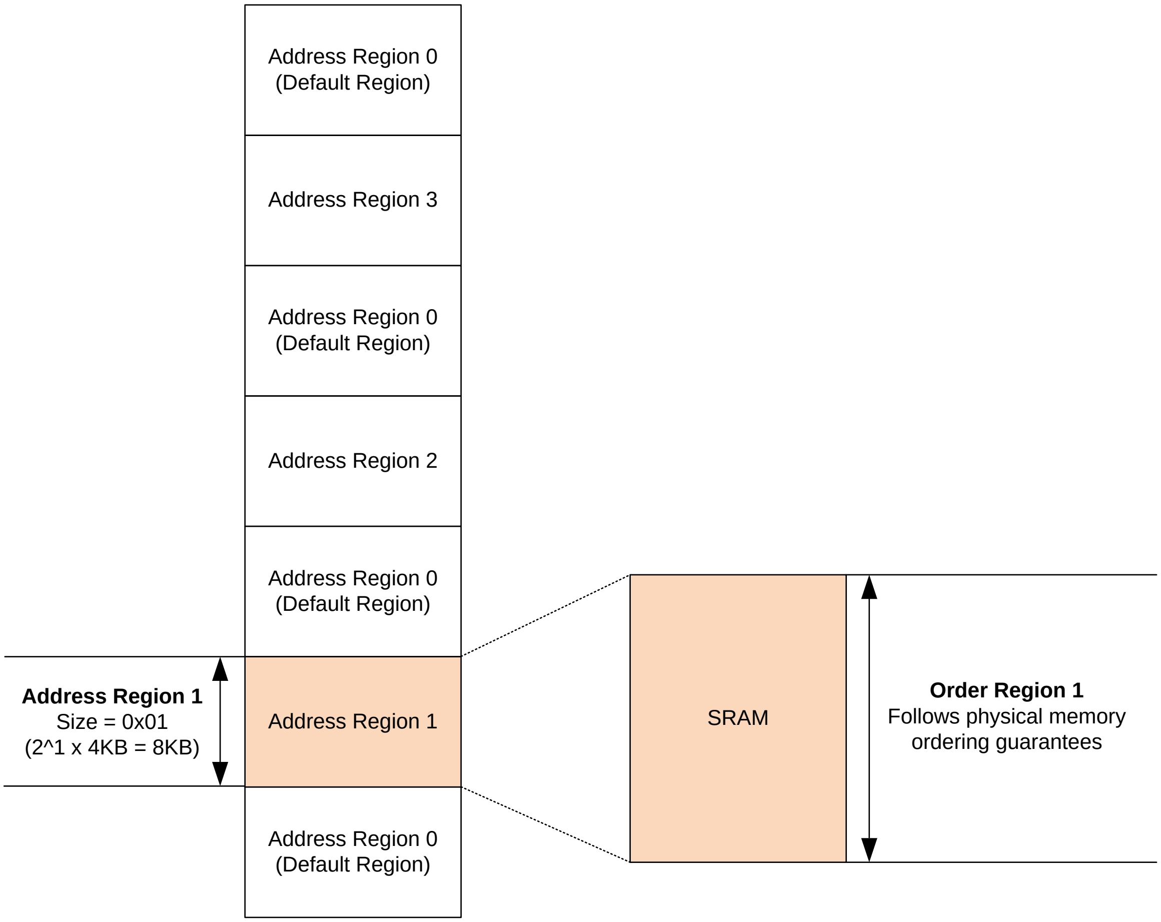
Table 4-34: Example por_hni_sam_addrregion0_cfg register programming

Bits	Field name	Configured value
[62]	pos_early_wr_comp_en	1'b1
[61]	pos_early_rdack_en	1'b1
[60]	ser_devne_wr	1'b0
[59]	ser_all_wr	1'b0
[58]	physical_mem_en	1'b0
[5:0]	order_reg_size	6'h4

Address region 1

In this example, the whole of address region 1 is assigned to an 8KB SRAM region. The following figure shows the example configuration for address region 1.

Figure 4-25: Example address region 1 configuration



Address region 1 starts at base address 0x0000_0000_2000. The whole of address region 1 is considered as one order region, which is 8KB in size. However, because there is SRAM behind this region, it is mapped as physical memory. Therefore, ordering is only maintained between all overlapping requests to a single 64B cache line. In other words, accesses to different cache lines are not kept in order.

The following table shows the configured values for the por_hni_sam_addrregion1_cfg register.

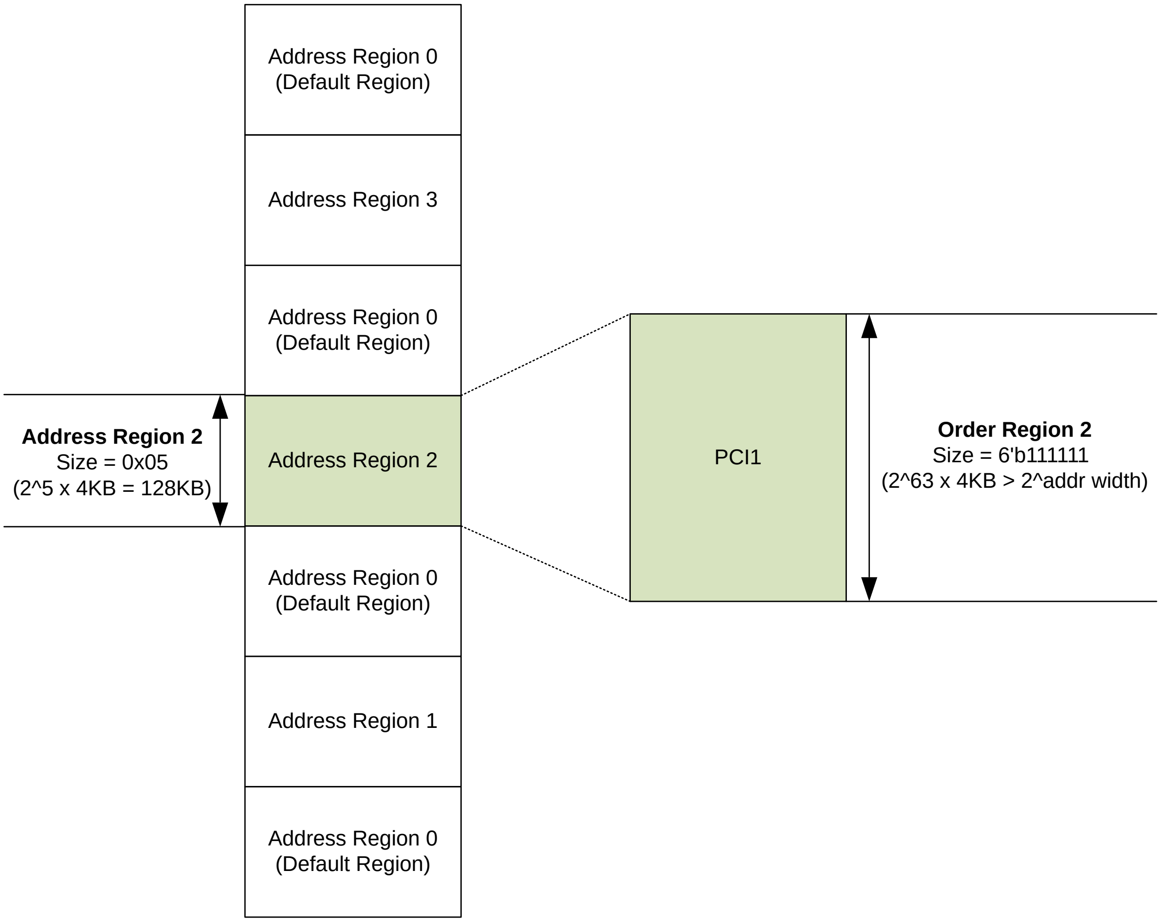
Table 4-35: Example por_hni_sam_addrregion1_cfg register programming

Bits	Field name	Configured value
[63]	valid	1'b1
[62]	pos_early_wr_comp_en	1'b1
[61]	pos_early_rdack_en	1'b1
[60]	ser_devne_wr	1'b0
[59]	ser_all_wr	1'b0
[58]	physical_mem_en	1'b1
[55:16]	base_addr	40'h0000_0000_2
[15:10]	addr_region_size	6'h1
[5:0]	order_reg_size	6'h1

Address region 2

In this example, address region 2 is assigned to a PCI1 region. The following figure shows the example configuration for address region 2.

Figure 4-26: Example address region 2 configuration



Address region 2 starts at base address 0x0000_0002_0000 and is 128KB in size. The order region 2 size is configured to the maximum value, 6'b111111, or 2^{63} × 4KB. Therefore, 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 por_hni_sam_addrregion2_cfg register.

Table 4-36: Example por_hni_sam_addrregion2_cfg register programming

Bits	Field name	Configured value
[63]	valid	1'b1
[62]	pos_early_wr_comp_en	1'b1
[61]	pos_early_rdack_en	1'b1
[60]	ser_devne_wr	1'b0
[59]	ser_all_wr	1'b0
[58]	physical_mem_en	1'b0

Bits	Field name	Configured value
[55:16]	base_addr	40'h0000_0002_0
[15:10]	addr_region_size	6'h5
[5:0]	order_reg_size	6'b111111

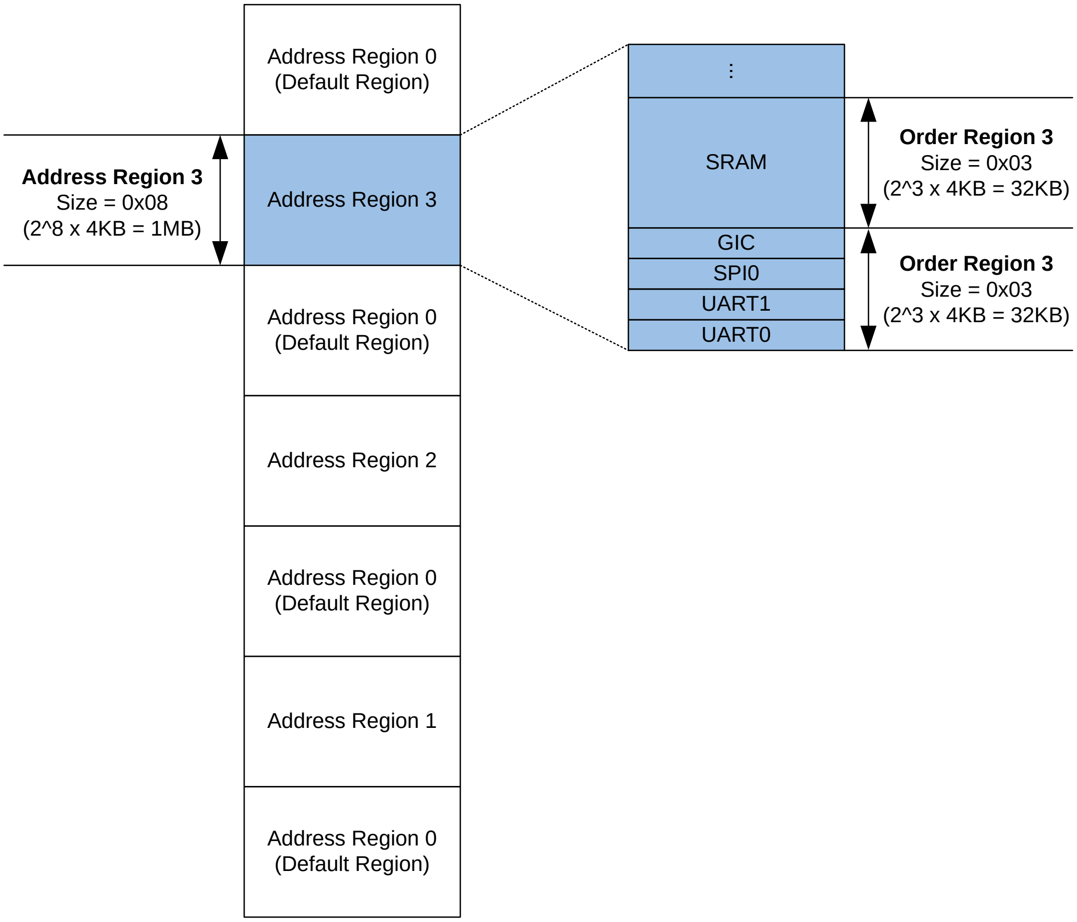
Address region 3

In this example, address region 3 is mapped to the following regions:

- UART0
- UART1
- SPI0
- GIC
- SRAM

The following figure shows the example configuration for address region 3.

Figure 4-27: Example address region 3 configuration



Address region 3 starts at base address 0x0000_0020_0000 and is 1MB in size. The order region 3 size is set to 32KB, which is less than the address region 3 size of 1MB. Therefore, a total of 32 order regions of 32KB each are contained within address region 3. UART0, UART1, SPI0, andGIC regions map to one order region, so all requests to these peripherals are ordered. SRAM also maps to a single order region, so all requests to SRAM are ordered. SRAM maps to a separate order region from UART0, UART1, SPI0, and GIC. Therefore, requests to SRAM are not ordered with respect to requests to UART0, UART1, SPI0, and GIC, and the other way around.

The following table shows the configured values for the por_hni_sam_addrregion3_cfg register.

Table 4-37: Example por_hni_sam_addrregion3_cfg register programming

Bits	Field name	Configured value
[63]	valid	1'b1
[62]	pos_early_wr_comp_en	1'b1
[61]	pos_early_rdack_en	1'b1

Bits	Field name	Configured value
[60]	ser_devne_wr	1'b0
[59]	ser_all_wr	1'b0
[58]	physical_mem_en	1'b0
[55:16]	base_addr	40'h0000_0020_0
[15:10]	addr_region_size	6'h8
[5:0]	order_reg_size	6'h3

Note

In HN-P, HN-I 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.

4.10 Traffic flow functionality

CMN-650 has optional features to optimize the bandwidth and traffic flow of the mesh.

4.10.1 Dual DAT and RSP channels

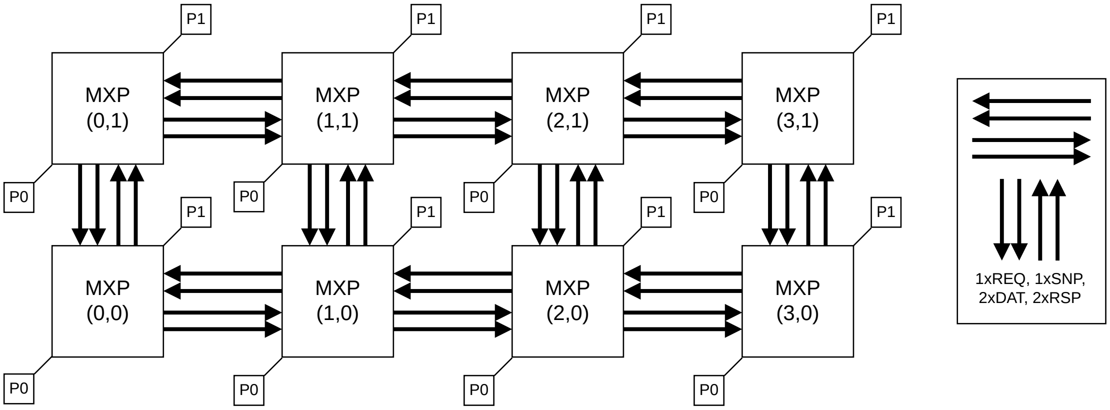
To reduce congestion within the mesh, CMN-650 provides a configuration option to double the number of DAT and RSP channels. This feature is useful if you have heavy traffic sources that saturate specific routing paths.

If the *2XDATRSP_EN* configurable parameter is set, the number of DAT and RSP channels per MXP-MXP connection increases from one to two. This option has the following benefits:

- Increases device upload bandwidth
- Increases MXP-MXP bandwidth

The following figure represents the dual DAT and RSP channels in the mesh.

Figure 4-28: Dual DAT and RSP channel topology



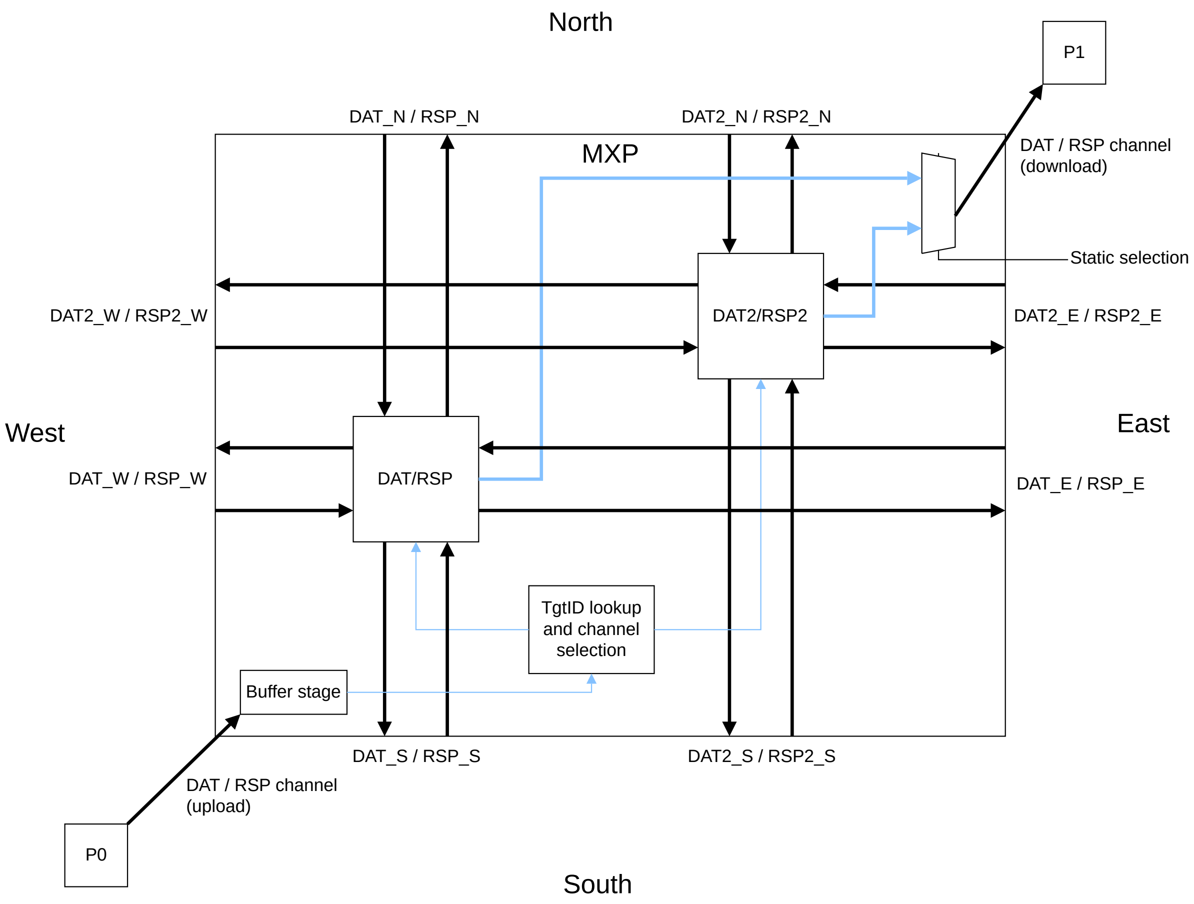
Boot-programmable registers determine which of the dual channels to select at flit upload. For more information about configuring the selection, see 4.10.3 Dual DAT/RSP channel selection registers on page 142. For the procedure to program these registers, see 5.4.4 Program the dual DAT/RSP channel selection scheme on page 843.

4.10.2 Dual DAT/RSP channel selection

If you enable the dual DAT/RSP channel feature, CMN-650 uses a TgtID-based selection scheme to determine which channel to upload DAT/RSP flits to. The interconnect also uses a static 2:1 multiplexer to download flits from the two channels.

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 both DAT and RSP channels.

Figure 4-29: Dual DAT/RSP 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 are mapped to one of the two RSP channels.

When you enable this feature by setting the *2XDATRSP_EN* configuration parameter, CMN-650 applies a default channel selection scheme. For mesh configurations where one of the mesh dimensions = 1 (1 × N configuration), traffic to all targets in the network use channel 0. 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. For more information, see 5.4.4 Program the dual DAT/RSP channel selection scheme on page 843.

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

Enabling this feature also adds a static 2:1 multiplexer for each channel type. By programming this feature, you select one channel for the target to use. The MXP uses this multiplexer to ensure that only flits from the correct channel are downloaded to the device.

4.10.3 Dual DAT/RSP channel selection registers

Boot-programmable registers define the TgtID-based LUT. The MXP uses this LUT to select DAT/RSP channels for flits.

If the dual DAT/RSP channel feature is enabled using por_mxp_multi_dat_rsp_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 TgtIDs, so you can configure the behavior for up to 64 targets overall.

After the dual DAT/RSP registers have been programmed, any remaining unprogrammed TgtIDs 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_dat_rsp_chn_sel_reg_*_valid bit of the register is set.

The following table shows the register format. Each 11-bit TgtID + CHN_SEL field specifies a TgtID and the DAT/RSP channel for flits with that TgtID.

Table 4-38: por_mxp_multi_dat_rsp_chn_sel_N (N=0-15) register format

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

Each 11-bit TgtID + 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_dat_rsp_chn_sel_reg_*_valid bit set. In this case:

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

The MXP also contains a control register, por_mxp_multi_dat_rsp_chn_ctrl. Program this register to indicate that the dual DAT/RSP channel selection configuration is complete.

The following table shows the control register format.

Table 4-39: por_mxp_multi_dat_rsp_chn_ctrl register format

Bit field	Description
[63:1]	Reserved

Bit field	Description
[0]	multi_dat_rsp_chn_sel_programmed

For more information about programming these registers, see 5.4.4 Program the dual DAT/RSP channel selection scheme on page 843.

4.10.4 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 master that is connected to and RN-I or RN-D does not block progress of traffic from another master.

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 configuration registers per port and an auxiliary control register:

- por_rn{i,d}_s_0-2_port_control
- por_rn{i,d}_aux_ctl

To enable this feature, you must program these registers.

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

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 4-40: Configuration register fields for dedicated GIC and MMU RN-I resources

Register name	Register field	Bits	Reset value	Description
por_rni_aux_ctl	dis_port_token	[17]	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	s0_rd_token	[17:11]	7'h0	Number of reserved read tokens for port 1, per slice
por_rni_s1_port_control	s0_wr_token	[24:18]	7'h0	Number of reserved write tokens for port 1, per slice
por_rni_s2_port_control	s0_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	s0_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_aux_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 4-6: 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.

Related information

- 5.3.1.7 por_rnd_aux_ctl on page 250
- 5.3.1.8 por_rnd_s0-2_port_control on page 252
- 5.3.13.7 por_rni_aux_ctl on page 637
- 5.3.13.8 por_rni_s0-2_port_control on page 639

4.10.5 Default XY routing behavior

By default, CMN-650 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.

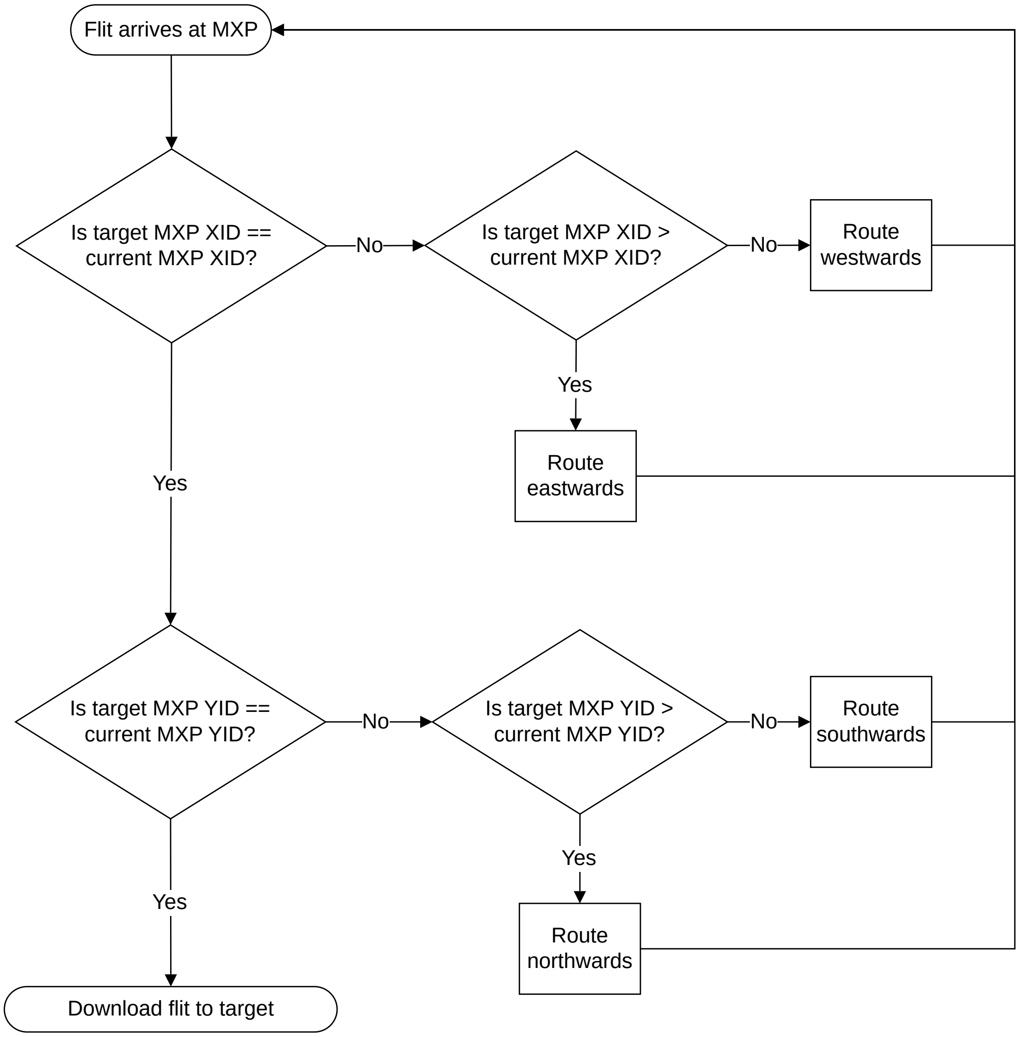
If the target MXP XID and the current MXP XID match, then the flit routing components are compared against the YID of the MXP. If YIDs do not match, then the MXP uses the following rule to decide the routing direction:

- 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 4-30: Default XY routing flow


Flit arrives at MXP
Is target MXP XID == current MXP XID?
No
Is target MXP XID > current MXP XID?
No
Route westwards
Yes
Route eastwards
Yes
Is target MXP YID == current MXP YID?
No
Is target MXP YID > current MXP YID?
No
Route southwards
Yes
Route northwards
Yes
Download flit to target

You can configure CMN-650 to override the default XY routing pattern for specific source-target pairs in the mesh. For more information about this feature, see 4.10.6 Non-XY routing on page 146.

4.10.6 Non-XY routing

You can configure up to 16 XP pairs in the CMN-650 mesh to route CHI traffic against the default XY routing algorithm. Non-XY routing improves the efficiency of traffic flow by reducing hotspots in the mesh layout.

By default, CMN-650 uses an XY routing mechanism to route flits through the mesh. For more information about the default XY routing mechanism, see 4.10.5 Default XY routing behavior on page 145.

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.

You 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.

Based on an identified hotspot, select one or two XPs for non-XY routing. The first XP is the point of XY route override, which overrides the XY route for the flit while still honoring the XY algorithm. The second XP is the point of YX turn where the flit is routed in the Y direction.

This feature only applies to the CHI channels, REQ, RSP, DAT, and SNP, not to the PUB channel.

4.10.7 Configuring non-XY routing behavior

A boot-programmable static *Lookup Table* (LUT) in each XP controls non-XY routing.

You configure support for this feature by setting the *XY_OVERRIDE_CNT* parameter. For more information, see 4.10.6 Non-XY routing on page 146 and 2.4 Global configuration parameters on page 23.

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 4-41: 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.
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 4-42: por_mxp_xy_override_sel_* structure

Bit field	Name
[63]	VALID
[62:59]	Reserved
[58:48]	srcid_1
[47]	Reserved
[46:36]	tgtid_1
[35:34]	Reserved
[33]	yx_turn_enable_1
[32]	xy_override_enable_1
[31:27]	Reserved
[26:16]	srcid_0
[15]	Reserved
[14:4]	tgtid_0
[3:2]	Reserved
[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 5.4.5 Program non-XY routing registers on page 843.

4.10.8 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).

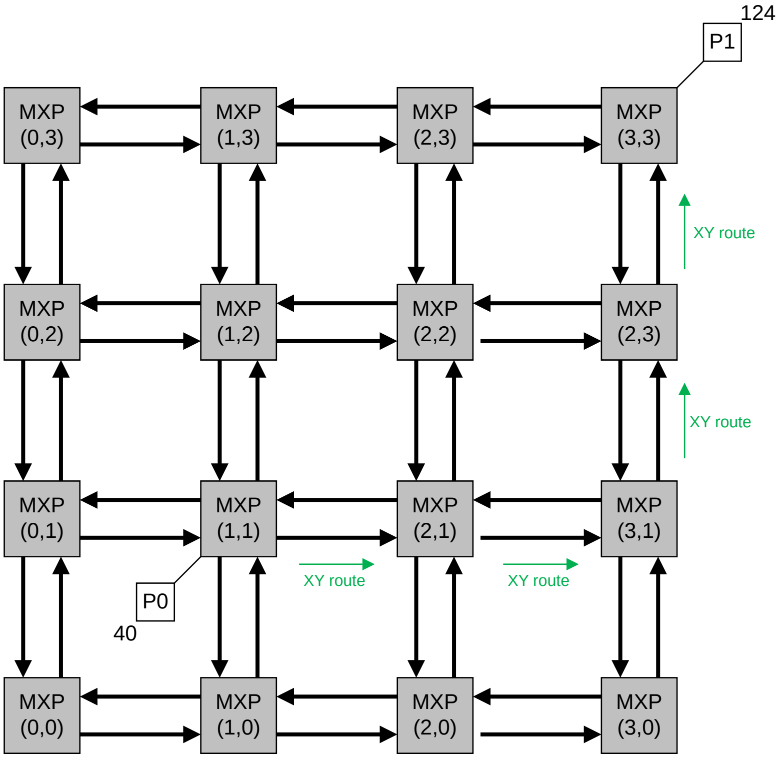
- If N → E turn is allowed at $XP_{xi,yi}$, then S → W turn is disallowed at every $XP_{xj,yj}$ where (xj > xi) and (yj < yi).
- If S → W turn is allowed at $XP_{xi,yi}$, then N → E turn is disallowed at every $XP_{xj,yj}$ where (xj < xi) and (yj > yi).

4.10.9 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 4x4 mesh configuration.

The following figure shows the default routing of the flit without non-XY routing.

Figure 4-31: Default XY routing example



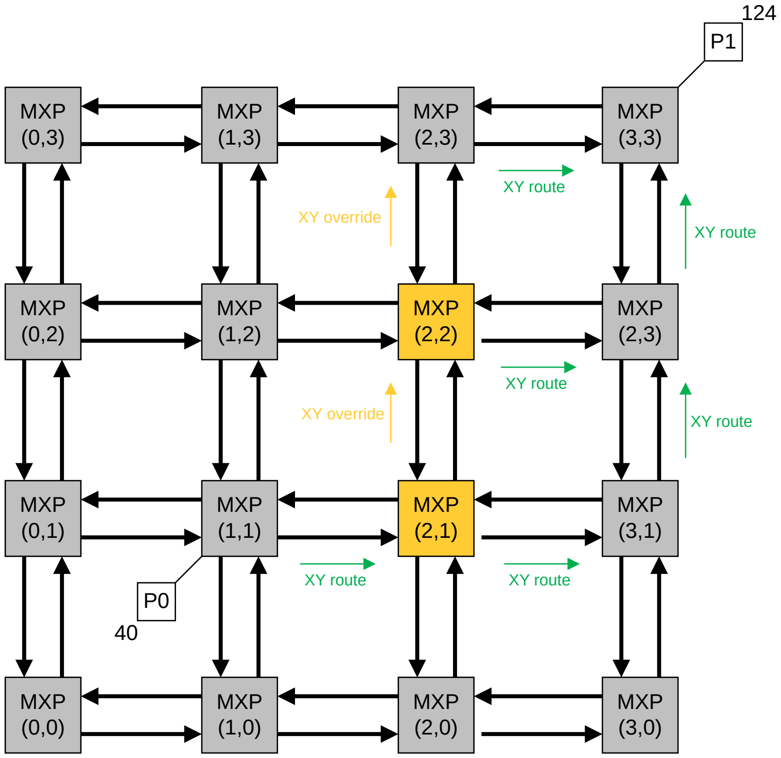
According to the standard XY routing algorithm, the flit follows the following route:

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 4-32: 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:

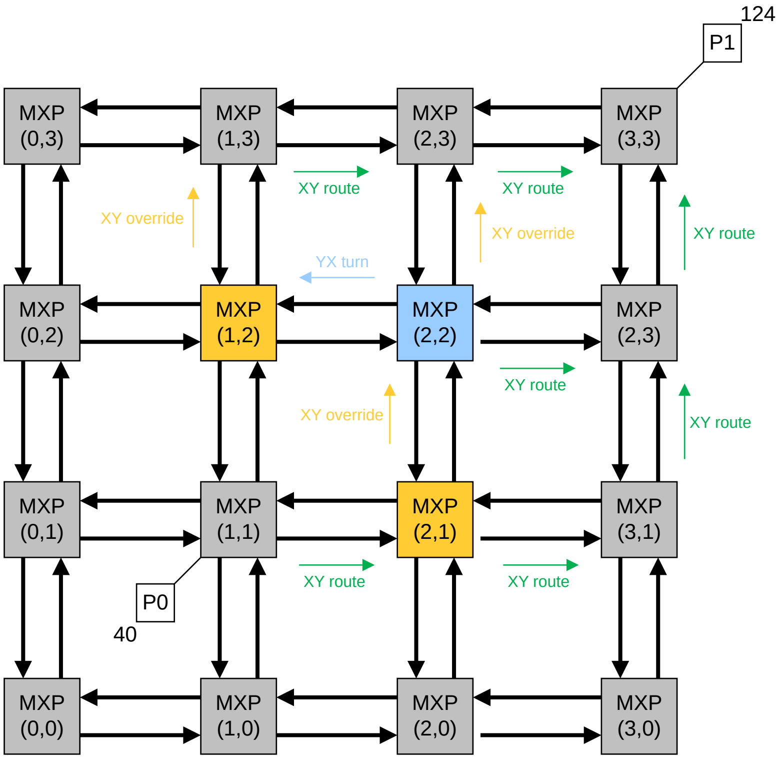
- If target MXP YID ≥ current MXP YID and a northern mesh port is present, then route northwards.
- If target MXP YID < current MXP YID and a southern mesh port is present, then route southwards.

- 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 4-33: 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.

4.11 CML functionality

CMN-650 supports CML and optional features to customize your CML configuration. When setting up a CML system, you must also be aware of the cross-chip routing mechanism to ensure that you program CMN-650 correctly.

4.11.1 Routing transactions across multi-chip systems

CMN-650 must be able to route transactions from one chip to the correct agent on another chip and maintain coherency across multiple chips. Therefore, CMN-650 must track local and remote resources by mapping remote IDs to local IDs on each chip. It uses several types of IDs and SAMs to determine the source and destination of transactions and converts between global and local IDs as necessary.

CMN-650 uses the following types of IDs to route transactions between different CML chips.

- CCIX *Requesting Agent ID* (RAID). RAIDs are global IDs, meaning that each RA across all CML chips must have a unique RAID.
- CCIX *Home Agent ID* (HAID). HAIDs are global IDs, meaning that each HA across all CML chips must have a unique HAID.
- LDID. LDIDs are local IDs and are assigned according to device type. Therefore, they must be unique only within a single device type on a single chip.

Only the CXG uses RAIDs for routing purposes. All other local and remote components that are visible to CMN-650 use and operate on sequentially assigned LDIDs. CXG devices bidirectionally map each CCIX RAID to an LDID.

Note

CMN-650 assigns default LDIDs at build-time. The default scheme assigns LDIDs from 0-n to local RNs of each type, sequentially. Remote RN-Fs must be assigned LDID values n+1 and above by the discovery software.
If your configuration does not use SF clustering, you can discover the default local LDID assignment by reading any of the following registers:

- HN-F por_hnf_rn_cluster<X>_physid registers
- CXRA por_cxg_ra_rnf_ldid_to_nodeid_reg registers

A few cycles after reset, the por_hnf_rn_cluster<X>_physid registers are prepopulated with the LDIDs for local RN-Fs within that chip. If your configuration does not use SF clustering, the LDIDs for local RN-Fs must not be changed.

If your configuration uses SF clustering, see 4.11.4 LDID assignment when using SF clustering on page 156.

CXRAs use an RA SAM to determine the target HAID for a request. CXHAs use an RN SAM to determine the target HN for an incoming request.

Related information

- 4.3.2 Logical Device IDs on page 92
- 6.2.11 Non-clustered and clustered mode for SF RN-F tracking on page 879
- 4.6 RA SAM on page 115
- 4.5 RN SAM on page 96

4.11.2 Mapping LDIDs to RAIDs in CXRA and CXHA

CMN-650 CXRA and CXHA nodes have programmable LUT registers to map local LDIDs to global RAIDs for all remote RAs in the system. These registers are used to determine transaction sources and targets during cross-chip routing.

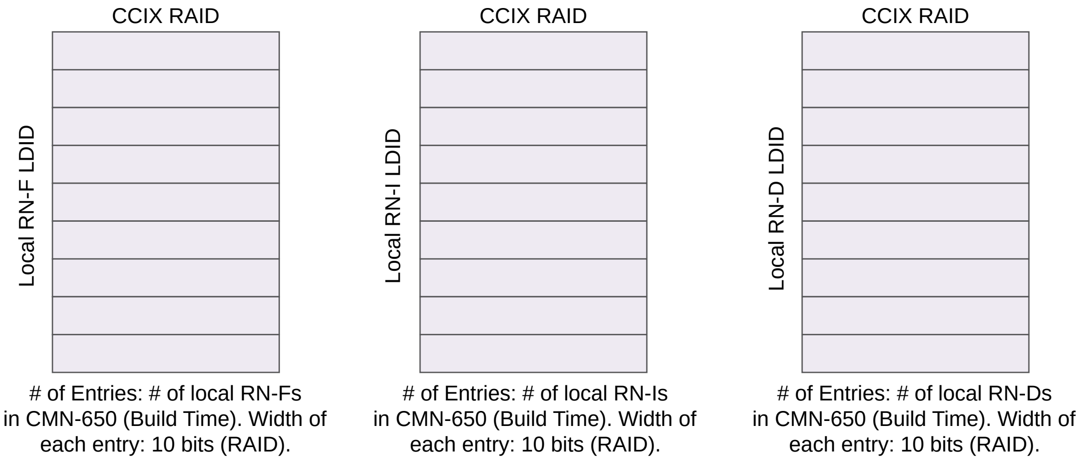
CXRA LUTs

CMN-650 CXRAs have the following programmable LUT registers which map LDIDs for each type of RN to their RAID value:

- por_cxg_ra_rni_ldid_to_exp_raid_reg_0-9
- por_cxg_ra_rnd_ldid_to_exp_raid_reg_0-9
- por_cxg_ra_rnf_ldid_to_exp_raid_reg_0-127

CMN-650 converts between these two values at various points when routing transactions from one chip to another. The following figure shows the format of these registers.

Figure 4-34: CXRA LDID to RAID LUT register format

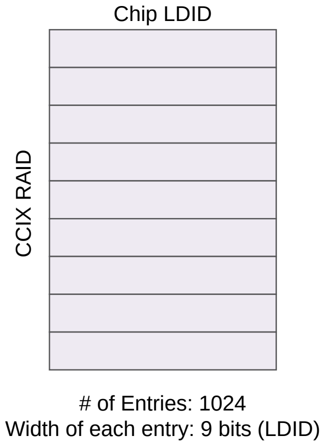

CCIX RAID
Local RN-F LDID
# of Entries: # of local RN-Fs in CMN-650 (Build Time). Width of each entry: 10 bits (RAID).
CCIX RAID
Local RN-I LDID
# of Entries: # of local RN-Is in CMN-650 (Build Time). Width of each entry: 10 bits (RAID).
CCIX RAID
Local RN-D LDID
# of Entries: # of local RN-Ds in CMN-650 (Build Time). Width of each entry: 10 bits (RAID).

The CCIX discovery software discovers all local RN-Fs, RN-Ds, and RN-Is, and programs their corresponding RAIDs into the RAID LUTs.

CXHA LUTs

CMN-650 CXHAs have programmable registers, por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255, which map some of or all RAIDs in the system to a local LDID value. If a remote RA can communicate with a local CHI HN, the CXHA on the local chip must have a register entry for that RA. The following figure shows the format of the registers that map RAIDs to LDIDs during CCIX discovery.

Figure 4-35: CXHA RAID to LDID LUT register format



Each entry in the register also contains a bit to indicate whether the remote RA is a caching agent, in other words an RN-F, or not. HN-Fs on a local chip use local LDIDs to track cache lines throughout the system in their SF. Therefore, each remote requesting caching agent in the system must have a unique LDID in these registers. If an RN-I and an RN-F have the same RAID, then you must only program the RN-F details in the entry. These unique LDIDs for remote RAs must not overlap with LDIDs that are assigned to local RN-Fs.

It is assumed that these LDIDs are programmed after CCIX discovery is complete. In other words, all the CXRAs have already been discovered and assigned an RAID.

Related information

- 5.3.2.11 por_cxg_ra_rni_ldid_to_exp_raid_reg0-9 on page 276
- 5.3.2.12 por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9 on page 277
- 5.3.2.13 por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127 on page 278
- 5.3.6.19 por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255 on page 413

4.11.3 Mapping SF LDID vector values to snoop targets in HN-F

The HN-F SF uses local LDID values that are associated with remote RAs to track cache lines that are cached in remote RAs. To route a snoop to an RN-F on another chip, the HN-F first sends a snoop to the correct local CXHA. The target node ID is identified using the LDID.

CMN-650 HN-Fs have the following programmable LUT registers to map SF LDIDs to the target CXHA:

- por_hnf_rn_cluster_0-63_physid_reg0
- por_hnf_rn_cluster_64-127_physid_reg0
- por_hnf_rn_cluster_0-127_physid_reg1
- por_hnf_rn_cluster_0-127_physid_reg2
- por_hnf_rn_cluster_0-127_physid_reg3

The following table shows an example of the programmed LUT in an HN-F.

Table 4-43: Example LDID to target node ID programming

Logical ID as index	HN-F programmable register	
	CHI node ID	ID valid
0	Local RN-F 0	1
1	Local RN-F 1	1
2	Local RN-F 2	1
…	…	…
7	Local RN-F 7	1
8	CXHA	1
9	CXHA	1
…	…	…
15	CXHA	1
16	Not programmed	0

Logical ID as index	HN-F programmable register	
	CHI node ID	ID valid
...	Not programmed	0
n	Not programmed	0

In the example shown in the preceding table, LDIDs 0-7 are assigned to local RN-Fs. LDIDs 8-15 are assigned to remote RN-Fs, and therefore the node ID of the relevant CXHA is associated with these entries.

The por_hnf_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-D), because CXHA is a proxy for all remote RN-Fs.

Related information

- 5.3.15.97 por_hnf_rn_cluster0-63_physid_reg0 on page 823
- 5.3.15.98 por_hnf_rn_cluster64-127_physid_reg0 on page 826
- 5.3.15.99 por_hnf_rn_cluster0-127_physid_reg1 on page 829
- 5.3.15.100 por_hnf_rn_cluster0-127_physid_reg2 on page 831
- 5.3.15.101 por_hnf_rn_cluster0-127_physid_reg3 on page 834

4.11.4 LDID assignment when using SF clustering

If your configuration has HN-F SF clustered mode enabled, you must program the LDIDs for local and remote RN-Fs in the system to meet clustering requirements.

If HN-F SF clustered mode is enabled, you are not required to use the default build-time LDID assignment, as 4.11.1 Routing transactions across multi-chip systems on page 152 describes. In clustered mode, local and remote RN-Fs can be assigned LDIDs from the full ID space to meet the clustering requirements.

To maximize SF efficiency, assigned LDIDs must be sequential, and the LDID space must not have any holes.

To assign LDIDs in clustered mode, program the local RN-F LDIDs in the por_mxp_p[0-1]_ldid_override register to match the clustering requirements at each RN-F port. In this mode, the local RN-F LDIDs are not pre-programmed in the por_hnf_rn_cluster<X>_physid registers of the HN-F out of reset. Therefore, you must explicitly program these registers to work with the clustering requirements.

The override LDID must be programmed in the por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg_[0-127] registers of the CXRA.

Related information

- 5.3.14.12 por_mxp_p0-1_ldid_override on page 662

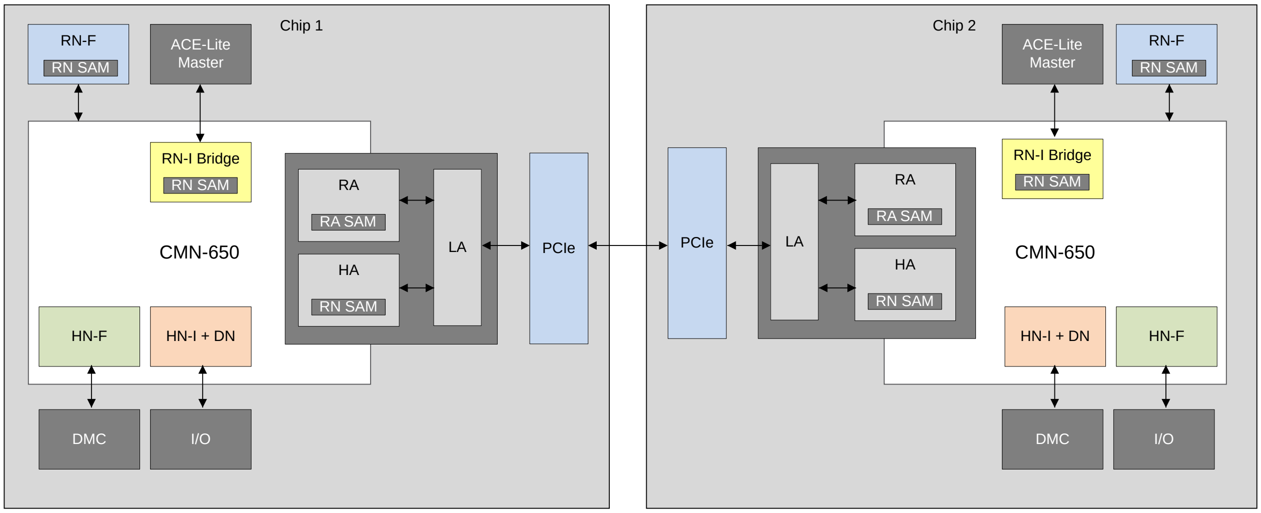
- 5.3.2.15 por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127 on page 281
- 6.2.12 Configuring clustered mode for SF tracking on page 881

4.11.5 Cross-chip routing examples

The CMN-650 CXRA and CXHA are the main components that route transactions from one chip to another through a CCIX link. To route from chip to chip and maintain coherency, the CXRA and CXHA use system-level ID values to track the RA of the transaction. When routing transactions within CMN-650, these ID values are converted to local ID values. We describe some example scenarios to demonstrate the routing and ID mapping as transactions are routed from chip to chip.

The following figure shows a block diagram of an example multi-chip system.

Figure 4-36: Example multi-chip system block diagram



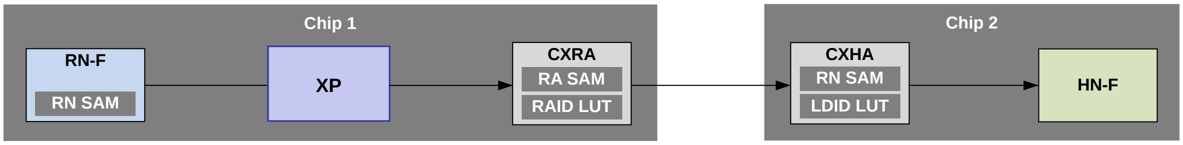
Note

The transaction examples that we use here are based on this example system. In the examples, we describe components according to the context of the chip that they are in. However, the same sequences can be used to describe cross-chip routing regardless of how many chips or devices there are in your system.

Routing request from local RN-F to remote HN-F

Consider a scenario where CMN-650 routes a request from an RN-F on chip 1 to an HN-F on chip 2. The following figure shows the path that the transaction takes.

Figure 4-37: RN-F to remote HN-F path and IDs



CHI SrcID = Node ID of the RN-F. Constant.

CHI TgtID = RN SAM output. Node ID of the CXRA. Programmable.

LDID = Logical ID of the RN-F. The XP populates this on every request.

CCIX SrcID = RAID LUT output. The LDID is used to get the RAID.

CCIX TgtID = RA SAM output. Address is used to get the HAID.

CCIX TxnID = CXRA Request Tracker ID.

CHI SrcID = Node ID of the CXHA. Constant.

CHI TgtID = RN SAM output. Programmable.

CHI TxnID = CXHA Request Tracker ID.

LDID = LDID LUT output. The RAID is used to get the LDID for each remote CXRA.

As the preceding figure shows, to get a transaction from a local RN-F to a remote HN-F, the following sequence of events happens:

1. The RN-F on chip 1 populates the SrcID of the transaction with its own node ID. It also uses the RN SAM to look up the CHI target ID for requests to that PA. The RN SAM returns the node ID of a CXRA node, which is populated as the TgtID.
2. The XP populates the LDID of the requesting RN-F on the request.
3. The CXRA uses its RAID to LDID LUT to map the LDID of the incoming request to the global RAID value for the requesting RN. The CXRA uses the RAID value as the SrcID for the outgoing CCIX request. To determine the TgtID, the CXRA uses the RA SAM to look up the HAID for the target CXHA. The CXRA request tracker ID determines the TxnID for the request.
4. The CXHA on chip 2 generates a local request and populates the CHI SrcID of the request with its own node ID. It uses the RN SAM to look up the CHI TgtID for requests to that PA. It populates the CHI TxnID with the CXHA request tracker ID. It also populates a local LDID value. This LDID is determined according to the SrcID of the incoming request, which is the RAID of the remote requestor. The CXHA maps this RAID value to a local LDID using the RAID to LDID LUT registers.
5. The HN-F receives the request.

The HN-F SF has a logical ID vector tracking resource. The LDID is passed to HN-F in all CHI REQ flits that the HN-F receives. For transactions from remote RN-Fs, the HN-F uses the LDID that is populated in step 4 as the logical ID for SF tracking purposes. If you have not enabled SF clustered mode, the SF tracks LDIDs individually. If SF clustered mode is enabled, then multiple LDIDs are aliased to a single logical ID.

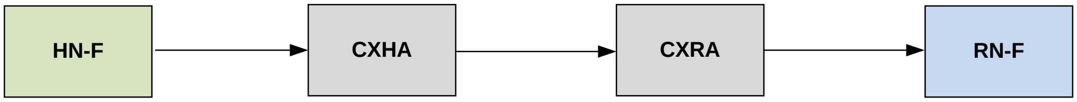
The total number of bits in the SF vector is calculated based on configuration parameters. You can also make the vector larger using the *SF_RN_ADD_VECTOR_WIDTH* configuration parameter.

For a transaction originating at an RN-I or RN-D, the sequence is similar. However, since RN-I and RN-D nodes are internal to CMN-650, their LDID is assigned during CMN-650 generation. Therefore, they send the LDID on every request, meaning that step 2 is not necessary.

Routing response from remote HN-F to local RN-F

The following figure shows the path to route a response from the HN-F on chip 2 to the RN-F on chip 1.

Figure 4-38: Remote HN-F to RN-F path and IDs



CHI SrcID = NodeID of HN-F (Constant).

CHI TgtID = SrcID from request (CXHA NodeID).

CHI TxnID = TxnID from request.

CCIX SrcID = CXHA HAID (Programmable).

CCIX TgtID = From incoming request.

CCIX TxnID = From incoming request.

CHI SrcID = NodeID of CXRA (Constant).

CHI TgtID = From incoming CHI request.

CHI TxnID = From incoming CHI request.

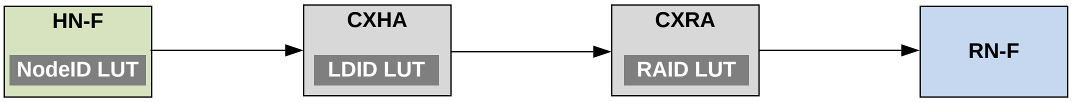
As the preceding figure shows, to get a response from a local HN-F to a remote RN-F, the following sequence of events happens:

1. The HN-F on chip 2 generates a CHI response transaction with the SrcID as its own node ID. The TgtID that the HN-F uses is the SrcID of the request, which is the CXHA node ID. The TxnID is the TxnID of the received request.
2. The CXHA generates a CCIX response transaction with the SrcID as its own HAID. It generates the TgtID and TxnID according to the original request.
3. The CXRA on chip 1 generates a CHI response transaction with the SrcID as its own node ID. It populates the TgtID and TxnID according to the original request.
4. The RN-F on chip 1 receives the response.

Routing a snoop from a local HN-F to a remote RN-F

Consider a scenario where the incoming request to the HN-F at chip 2 results in a snoop being sent to an RN-F on chip 1. The following figure shows the path that the snoop must take.

Figure 4-39: Snoop from HN-F to remote RN-F path and IDs



CHI SrcID = NodeID of HN-F (Constant).

CHI TxnID = HN-F Tracker ID.

LDID = from Snoop Filter.

CCIX SrcID = CXHA HAID (Programmable).

CCIX TgtID = O/P of LDID LUT. Incoming LDID is compared against LDID LUT to get the RAID.

CCIX TxnID = CXHA Request Tracker ID.

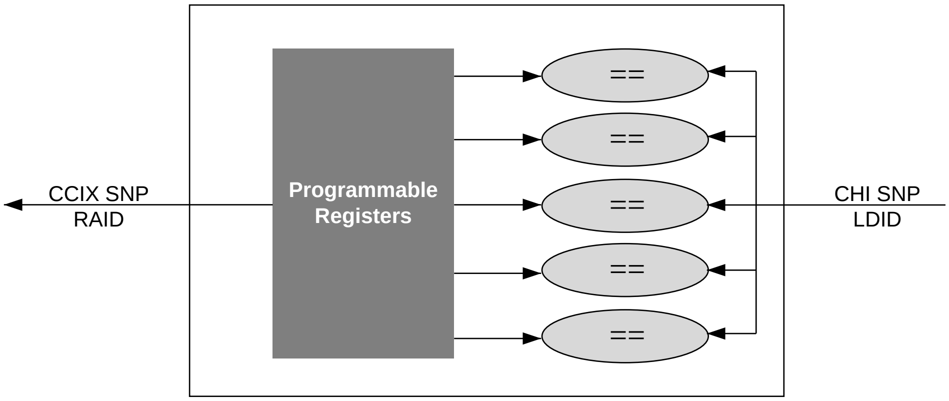
CHI SrcID = NodeID of CXRA (Constant).

As the preceding figure shows, to get a snoop from a local HN-F to a remote RN-F, the following sequence of events happens:

1. The HN-F on chip 2 uses the LDID that is stored in the SF to look up the local target for the snoop. To find the target, it looks up the node ID that is associated with the SF LDID in its LUT. This node ID corresponds to the CXHA node that is associated with the remote RN-F. It then generates a CHI snoop transaction with the TgtID as the CXHA node ID. It populates the SrcID as its own node ID, the TxnID as the HN-F tracker ID, and the LDID as the value from the SF.
2. The CXHA uses the LDID of the incoming request to perform a content match against its RAID to LDID LUT. It then generates a CCIX snoop with the TgtID set as the output of the RAID to LDID LUT. It populates the SrcID as its own HAID, and the TxnID as the CXHA request tracker ID.
3. The CXRA on chip 1 uses the TgtID of the incoming CCIX SNP to match against its RAID to LDID LUT. The LDID is then compared against a static list of RN-F node IDs. The CXRA generates a CHI snoop with the TgtID set to the node ID that corresponds to the LUT output. It populates the SrcID as its own node ID and the TxnID as the CXRA request tracker ID.

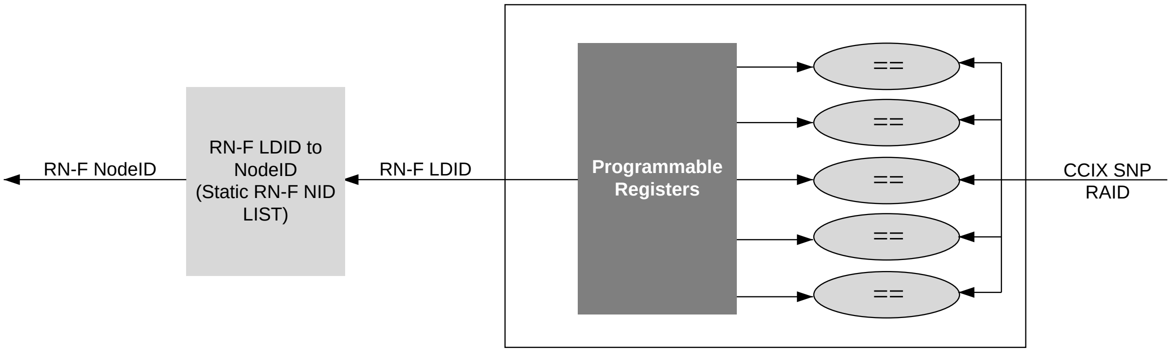
The following figure shows the content match logic that the CXHA uses to derive the RAID.

Figure 4-40: CHI SNP LDID to CCIX SNP RAID flow



The following figure shows the detailed flow of a CHI SNP LDID to CCIX SNP RAID conversion.

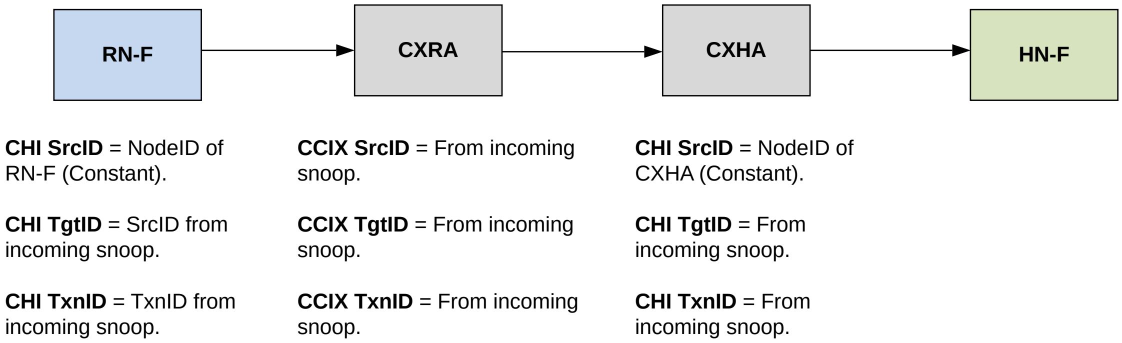
Figure 4-41: CCIX SNP RAID to CHI RN-F LDID flow



Routing a snoop response from a remote RN-F to a local HN-F

The following figure shows the path that the snoop response takes from an RN-F on chip 1 to an HN-F on chip 2.

Figure 4-42: Remote RN-F to HN-F with all IDs generated


RN-F
CXRA
CXHA
HN-F
CHI SrcID = NodeID of RN-F (Constant).
CHI TgtID = SrcID from incoming snoop.
CHI TxnID = TxnID from incoming snoop.
CCIX SrcID = From incoming snoop.
CCIX TgtID = From incoming snoop.
CCIX TxnID = From incoming snoop.
CHI SrcID = NodeID of CXHA (Constant).
CHI TgtID = From incoming snoop.
CHI TxnID = From incoming snoop.

As the preceding figure shows, to get a snoop response from a local RN-F to a remote HN-F, the following sequence of events happens:

1. The RN-F on chip 1 generates a CHI snoop response with the SrcID set to its own node ID. It generates the TgtID and TxnID based on the received snoop.
2. The CXRA generates a CCIX response with the SrcID, TgtID, and TxnID set according to the received snoop.
3. The CXHA generates a CHI response with the SrcID set to its own node ID, and the TgtID and TxnID set according to the received snoop.
4. The HN-F receives the snoop response.

Related information

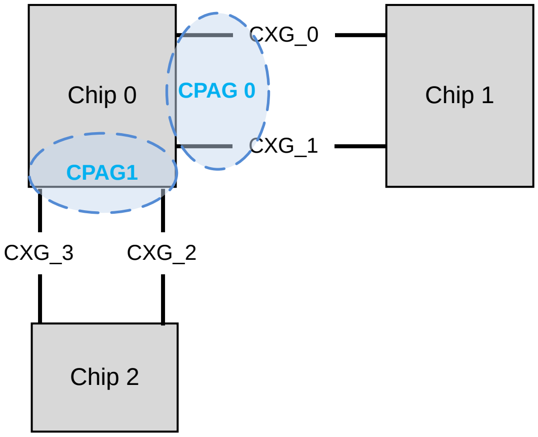
- 4.11.2 Mapping LDIDs to RAIDs in CXRA and CXHA on page 153
- 4.11.3 Mapping SF LDID vector values to snoop targets in HN-F on page 155
- 6.2.10 RN-F tracking in the SF on page 878

4.11.6 CCIX Port Aggregation Groups

The CMN-650 CML configuration supports up to ten CXGs. These CXGs can be grouped in up to five *CCIX 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 CXGs that are grouped into two CPAGs.

Figure 4-43: CCIX Port Aggregation Groups



In the preceding example, the following CPAGs are present:

CPAG_0 CPAG with two CCIX ports, CXG_0 and CXG_1, to connect chip 1 to chip 0.
CPAG_1 CPAG with two CCIX ports, CXG_2 and CXG_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.

4.11.7 CML Symmetric Multiprocessor support

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

CMN-650 supports SMP systems if the systems were built using the same version of CMN-650.

When set, the provided SMP mode option enables DVM, GIC-D, Exclusives, MPAM, and processor event communication across CCIX links using a micro-architected mechanism.

The microarchitecture support for propagating Trace Tag across CCIX links is known as Remote Trace Tag. Remote Trace Tag is only enabled in SMP mode. The sender CXG only propagates Remote Trace Tag on the outgoing CCIX request, and the receiving CXG only propagates Remote Trace Tag from the incoming CCIX request. This feature ensures all subsequent CCIX messages that are part of the same transaction use the same CCIX TxnID. Similarly, the sender CXG only propagates Remote Trace Tag on the outgoing CCIX Snoop, and the receiving CXG only from the incoming CCIX snoop.

For more information about programming CMN-650 for SMP mode, see 5.5 CML programming on page 849.

4.11.8 CML CCIX Slave Agent support

CML supports CCIX-independent memory expansion where the CCIX link is used to communicate only with the Slave Agent on the remote chip.

For more information, see the *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*.

To enable this support, set the cxsa_mode_en bit in the por_cxg_ra_cfg_ctl register. See 5.3.2.5 por_cxg_ra_cfg_ctl on page 268.

In this mode, CXRA accepts ReadNoSnp, WriteNoSnp, and CMO requests from HN-Fs and sends them to the Slave Agent on the remote chip. An HN-F SAM range-based memory region is programmed with both of the following items:

- The address range of the remote Slave Agent.
- The node ID of the corresponding CXG block that communicates with that remote Slave Agent.

The RA SAM inside CXRA is programmed with the CCIX Source ID (HAID) and Target ID (SAID) which are used in the CCIX header.

For further CXSA programming requirements, see 5.5 CML programming on page 849.

4.11.9 CXHA passive buffer support

CMN-650 supports an optional passive buffer in CXHAs. This buffer lets the CXHA give extra CCIX Request and Data credits.

The CXHA passive buffer is present if the CXHA *HA_PASS_BUFF_DEPTH* parameter is set to a nonzero value.

CMN-650 uses RAMs to implement the data part of the passive buffer. This part of the passive buffer contains the following items:

- 64B of data
- Corresponding *Byte Enables* (BEs)
- Poison
- Any metadata that is associated with the data

CCIX CopyBack requests cannot be allocated in the passive buffer. Therefore, request and data credits from the CXHA active tracker are granted to service them. By default, CXHA reserves 64 CCIX Request and Data credits in total from the active tracker to service these CopyBack requests. However, you can configure this number to suit the requirements of your system, by programming the num_copyback_crds field of the por_cxg_ha_aux_ctl register. This field can take any value between the number of enabled SMP links to one less than the size of the write data buffer.

Note

The size of the write data buffer is equal to the value of the *HA_NUM_WRBUF* parameter.

When programming the value of num_copyback_crds, you must ensure that enough active buffer entries are available to service other non-CopyBack write requests.

If the passive buffer is present but any of the CCIX protocol links at the CXHA are configured to operate in non-SMP mode, then the passive buffer is disabled and bypassed. In this case, all CCIX request and data credits are granted from the CXHA active tracker. The disable_passive_buf field in the por_cxg_ha_aux_ctl register enables the passive buffer bypass mode during initial boot programming.

The CXRA tracks reserved CopyBack credits from the remote CXHA separately and uses them to send CopyBack requests. CXRA detects the number of reserved CopyBack credits during the initial link bring up process. Usually, this hardware detection mechanism is sufficient for a CXRA and CXHA pair to detect passive buffer mode and usage of special credits. However, if both of the following conditions are met, you can explicitly enable reserved credit usage during initial boot programming:

- It is known upfront that a CXHA has a passive buffer.
- All links of the CXHA are operating in SMP mode.

If these conditions are met, then software can use the lnk<X>_spcl_cbkwr_crd_en field in the por_cxg_ra_cxprtcl_link<X>_ctl register at boot. This field explicitly enables reserved credit usage.

4.11.10 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 por_cxg_ra_cfg_ctl register.

This requirement applies to each enabled CCIX link at a given CCIX port. For example, by default all the reservations are enabled in SMP mode. Therefore, a minimum of four request and four data credits must be granted per CCIX link. These credits are used by certain traffic types, such as QoS-15, to progress in a loaded system. For more information, see 5.3.2.5 por_cxg_ra_cfg_ctl on page 268.

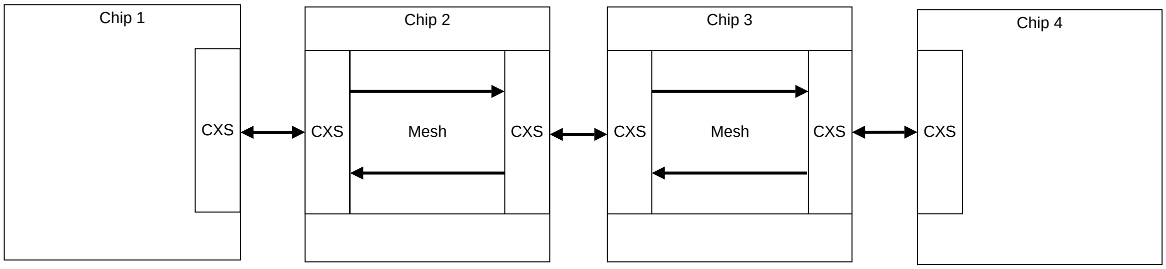
4.11.11 CML port-to-port forwarding

CMN-650 supports CML configurations with multiple CXS connections. CMN-650 can pass transactions from one CXS connection to another through the mesh, eliminating the requirement for external switching logic.

The port-to-port forwarding feature allows the CMN-650 mesh to act as a bridge between two CCIX chips. Using this feature, you can create various multichip topologies using only CCIX links, including a CCIX daisy-chain and CCIX mesh.

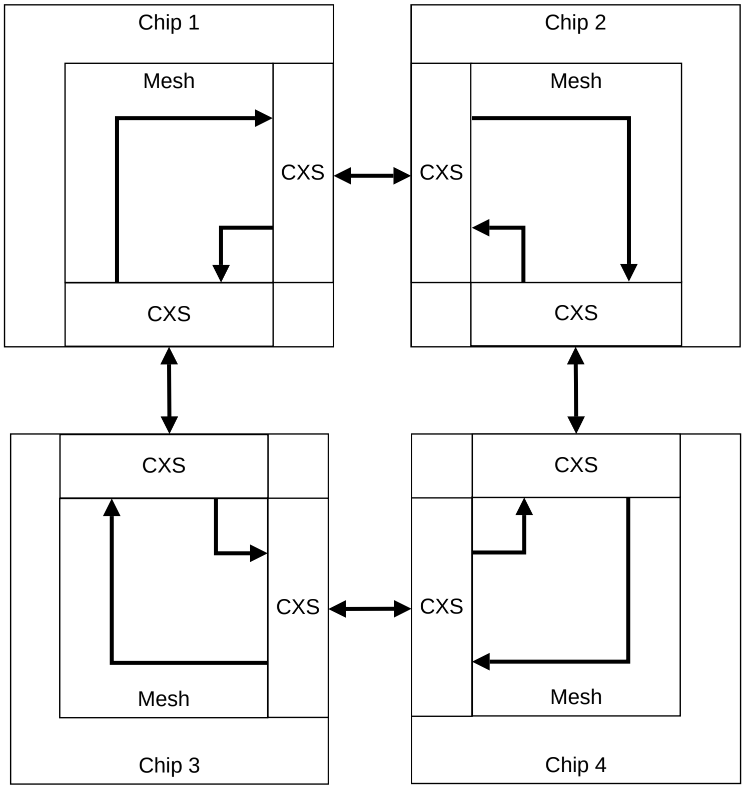
The following figure shows a daisy-chain topology using port-to-port forwarding. Each chip can target traffic to every other chip in the chain.

Figure 4-44: Four chip CCIX daisy chain configuration using port-to-port forwarding



The following figure shows a CCIX mesh topology using port-to-port forwarding. Each chip can target traffic to every other chip in the CCIX mesh.

Figure 4-45: Four chip CCIX mesh configuration using port-to-port forwarding



For more information about programming the port-to-port forwarding feature, see 5.5.3 Program CML system to enable CCIX communication on page 853.

4.12 Discovery

Discovery is a software algorithm that is used to discover the configuration of CMN-650.

Software uses the discovery mechanism to identify the following properties:

- The CHI node ID and LDID corresponding to all node types

Note

CMN-650 has the following valid logical node types:

- DVM
- Global CFG
- DTC
- HN-F

- HN-I
- RN-D
- RN SAM
- RN-I
- SBSX
- XP

CMN-650 also has the following node types for CML functionality:

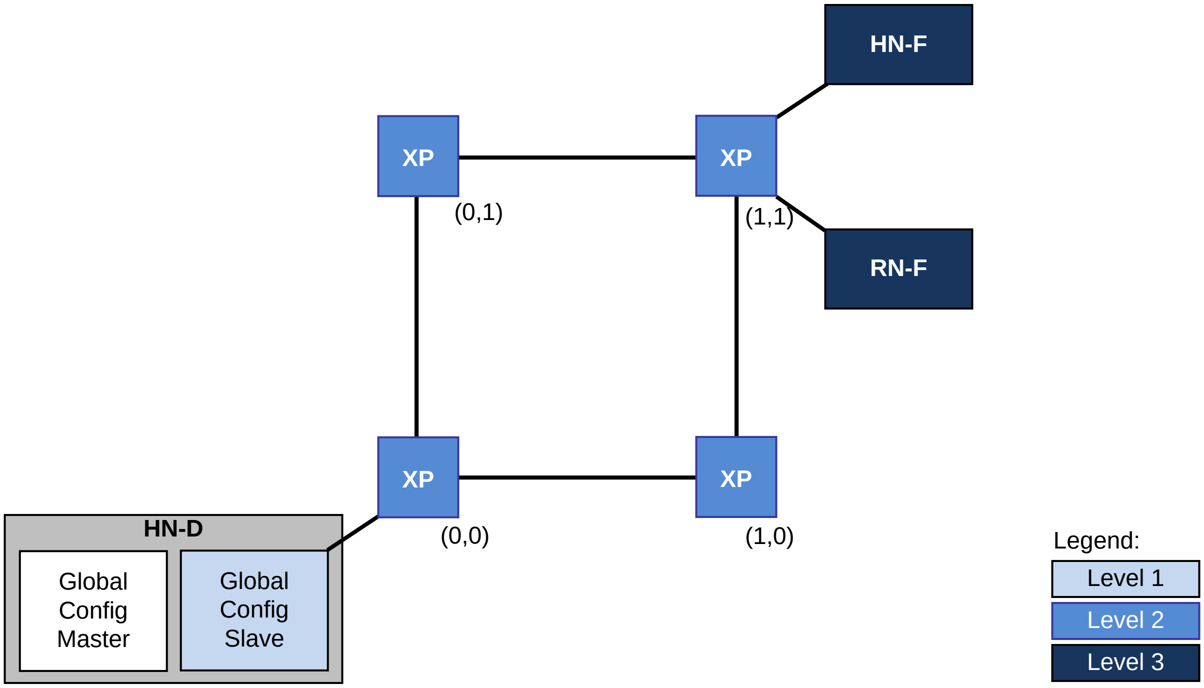
- CXRA
- CXHA
- CXLA

- Whether a discovered node is internal or external to CMN-650

The following figure shows an example configuration. In the example, after discovery, software has 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 4-46: 2 × 2 register tree example



4.12.1 Configuration address space organization

The way 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-650 configuration registers are mapped to. For a CMN-650 system where both the X and Y dimensions are eight or less:

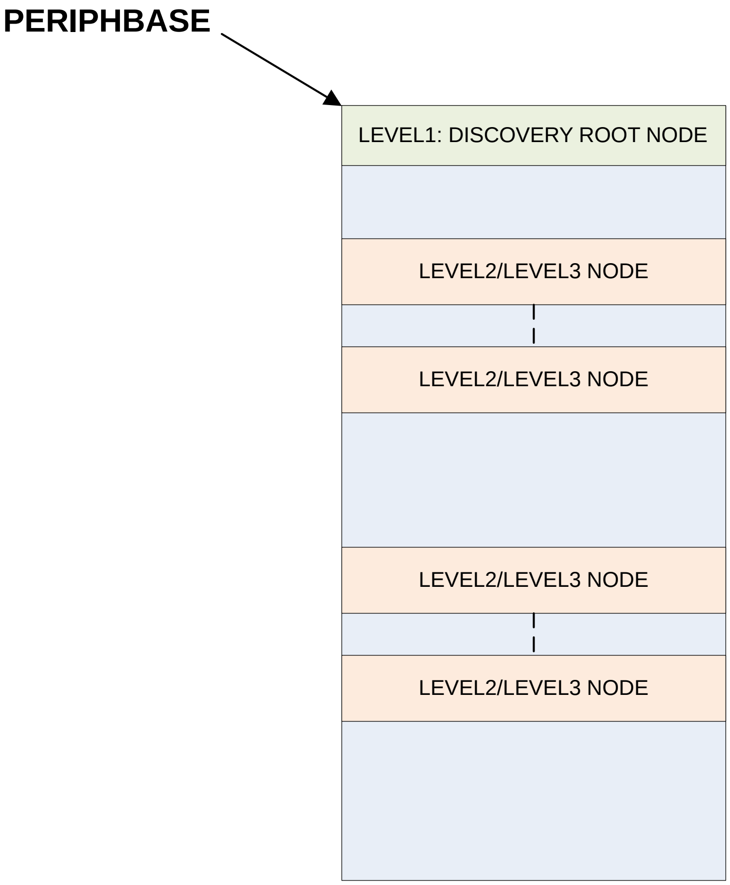
- This address must be aligned to 256MB.
- The maximum size of the address range is 256MB.

For a CMN-650 system where both the X and Y dimensions are nine or more:

- This address must be aligned to 1GB.
- The maximum size of the address range is 1GB.

Discovery determines specific addresses for individual system blocks that have **IMPLEMENTATION DEFINED** register spaces, as the following figure shows.

Figure 4-47: PERIPHBASE address map



CMN-650 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 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 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.

4.12.2 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.

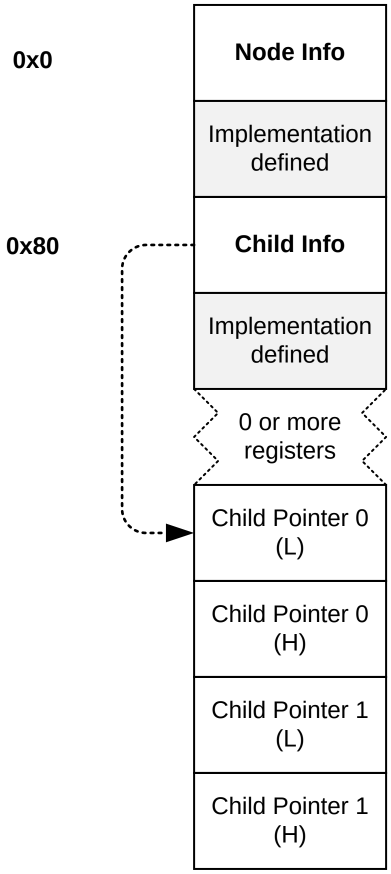
Child Information register
Indicates the child count and offset for the first register containing child node pointers. These optional Child Pointer registers each use 8B.

Note

The Node Information and Child Information registers are at fixed offsets for all nodes.

The following figure shows the basic node structure.

Figure 4-48: 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.

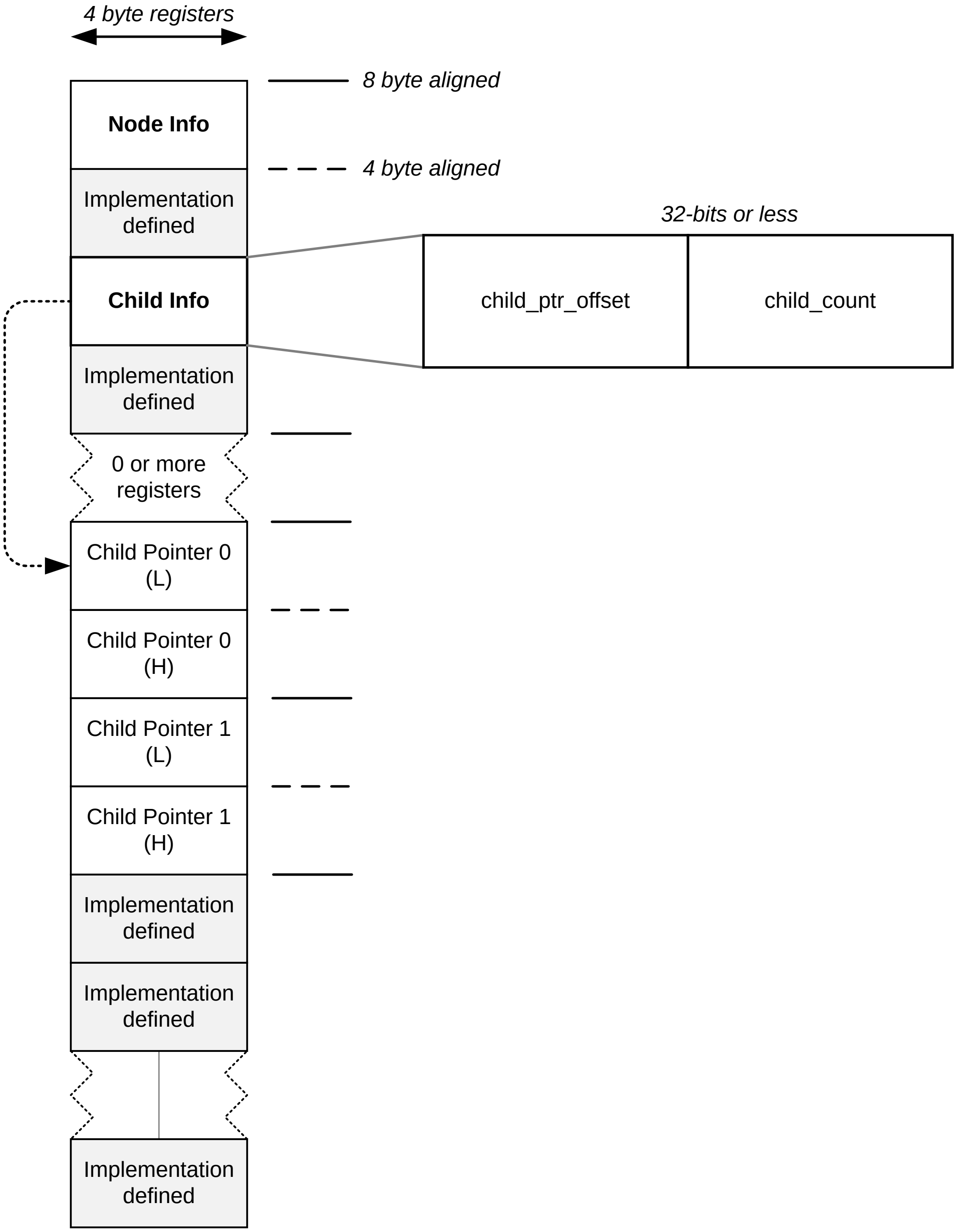
The child_ptr_offset field of the Child Information register indicates the Child Pointer 0 register offset, in bytes, from the Node Information register address.

Note

For a leaf node (node with no children), the child_count and child_ptr_offset fields must be set to zero.

The following figure provides the node structure detail.

Figure 4-49: Node structure detail



The following table shows the supported node types and the corresponding node_type values in the Node Information register.

Table 4-44: 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
MPAM_S	16'h0008
MPAM_NS	16'h0009
RN-I	16'h000A
RN-D	16'h000D
RN SAM	16'h000F
CXRA	16'h0100
CXHA	16'h0101
CXLA	16'h0102

4.12.3 Child pointers

Each child node has one child pointer register.

The address of the register containing the first child pointer is calculated using the following addresses and offsets:

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. For more information, see Figure 4-49: Node structure detail on page 173.

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 External Child Node bit of the child pointer register (bit[31]) has the following encodings:

1 Indicates that this CHILD POINTER is pointing to a Config Node that is external to CMN-650.

0 Indicates that this CHILD POINTER is pointing to a Config Node that is internal to CMN-650.

For CMN-650, external child nodes are only used for CXLA Config Node. 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

The device type corresponding to that child node helps the discovery software determine if the child node is RN-F, RN SAM, or CXLA. Every CXRH, CXHA, or CXRA node has a corresponding external CXLA node. Therefore, if the device type is CXRH, CXHA, or CXRA, then the external child node is CXLA. It is the responsibility of the discovery software to ensure that the external child node is powered ON before sending any config accesses to it.

Depending on the size of the mesh (X and Y dimensions), CMN-650 supports three different widths for encoding the X and Y dimension. The number of bits needed is selected based on the larger of the X and Y values.

Table 4-45: 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

4.12.4 Discovery tree structure

The one-time discovery process creates a lookup table that contains the addresses for all CMN-650 configured devices.

The discovery tree structure consists of three levels:

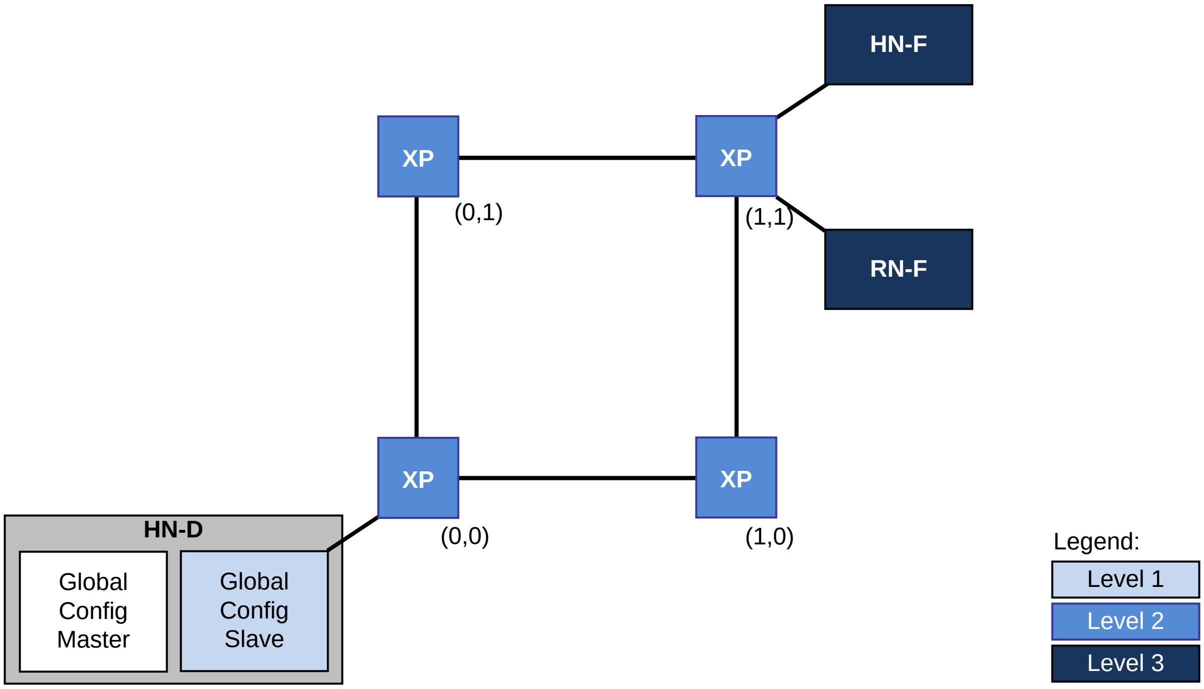
Level 1 Root Node, or the HN-D containing the Global Configuration Slave.

Level 2
XP layer.

Level 3
Leaf layer with one or two devices.

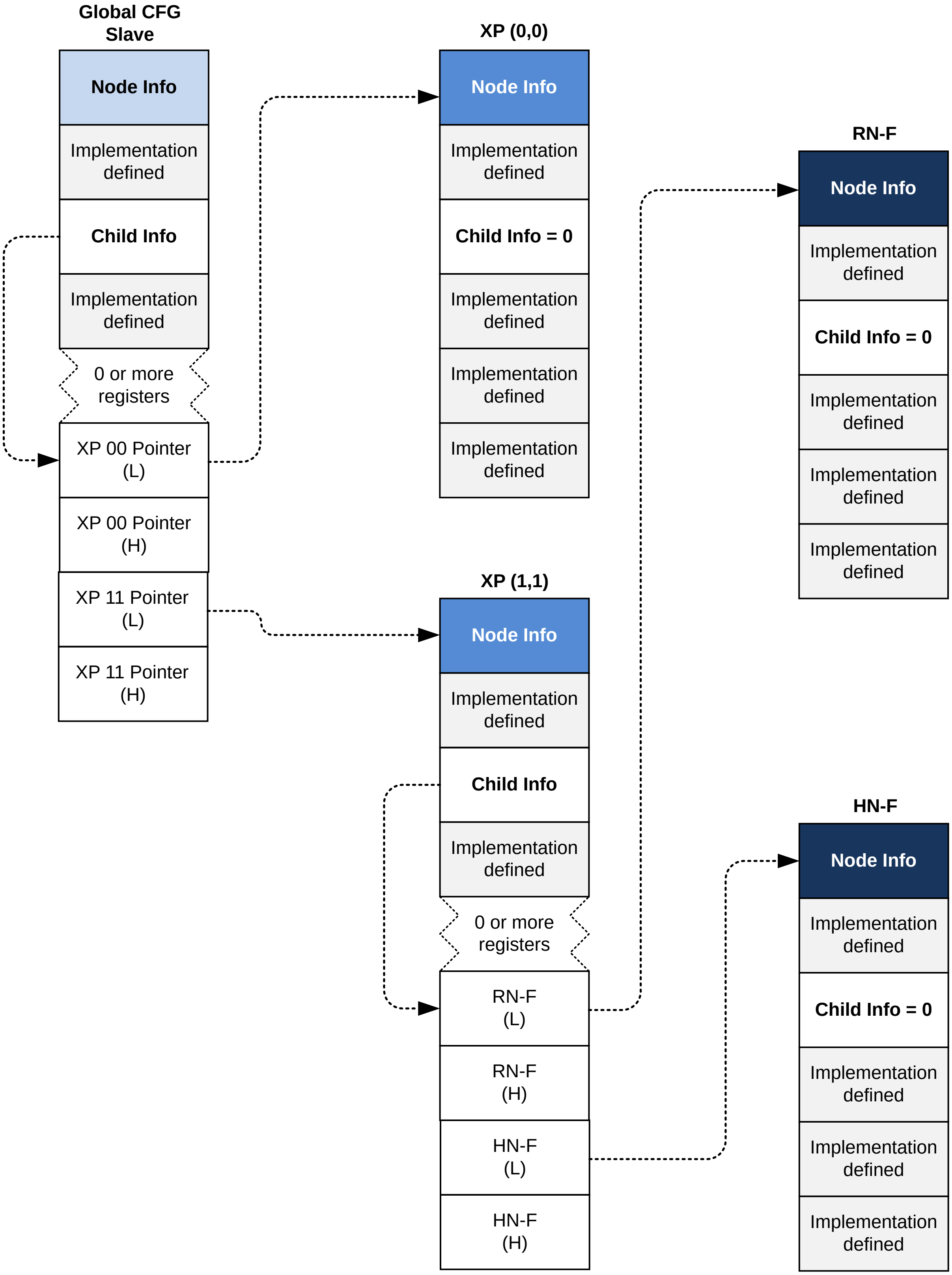
The following figure shows a 2 × 2 mesh configuration example with highlighted discovery tree levels.

Figure 4-50: 2 × 2 discovery tree example



The following figure shows the discovery tree structure for this 2 × 2 mesh configuration.

Figure 4-51: 2 × 2 discovery tree structure



4.13 Link layer

CMN-650 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. The transmitting device uses one credit per flit. The receiving device sends these credits back to the transmitting device, one at a time, after processing each flit. Subsequent flit transfers can then occur.

Note

This section refers to the credit roundtrip latency. This latency is measured in clock cycles, and is between:

1. The time a transmitting device uses a link layer credit to send a flit to the receiving device.
2. The earliest time when the transmitting device can receive that credit back from the receiving device and send a subsequent flit.

- A link deactivation mechanism. 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-650 is the receiving device. On flit download channels, CMN-650 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*.

4.13.1 Flit buffer sizing requirements

There are specific size requirements for the CMN-650 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 due to 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:

- 4.13.2 Flit uploads from RN-F or SN-F on page 179
- 4.13.3 Flit downloads at RN-F or SN-F on page 179

4.13.2 Flit uploads from RN-F or SN-F

For flit uploads, the *RXBUF_NUM_ENTRIES* parameter specifies the number of flit buffers in CMN-650.

For more information about *RXBUF_NUM_ENTRIES*, see 2.5 Device-level configuration parameters on page 25.

For optimal flit transfer bandwidth, this parameter must be set equal to the upload credit roundtrip latency, UpCrdLat<ch>. Use the following equation to calculate UpCrdLat<ch>:

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-650. This latency is measured in clock cycles between the following time points:
  1. The time that the RN-F or SN-F asserts the RX<ch>FLITV input for a flit that is uploaded to CMN-650.
  2. The earliest time when CMN-650 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 and SN-F interfaces, UpCrdLatInt<ch> = 1 on all CHI channels.
- UpCrdLatExt<ch> is the upload credit latency outside CMN-650. This latency is measured in clock cycles between the following time points:
  1. The time that CMN-650 asserts the RX<ch>LCRDV output when the credit is sent back to the RN-F or SN-F.
  2. 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.

4.13.3 Flit downloads at 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>.

Use the following equation to calculate DnCrdLat<ch>:

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-650. This latency is measured, in clock cycles, between the following time points:
    - The time that the RN-F or SN-F asserts the RX<ch>LCRDV input to CMN-650
    - The earliest time when CMN-650 asserts the RX<ch>FLITV output 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-650. This latency is measured, in clock cycles, between the following time points:
    - The time that CMN-650 asserts the RX<ch>FLITV output for a flit downloaded to the RN-F or SN-F
    - The earliest time when RX<ch>LCRDV input is asserted when the RN-F or SN-F returns the corresponding credit to CMN-650

4.14 Backward compatible RN-F support

CMN-650 is compliant with CHI-D, but can also contain RN-Fs that comply with CHI-B and CHI-C. Certain restrictions apply to how CMN-650 handles transactions that are sent from older RN-Fs to maintain backwards compatibility.

The following table shows how all CMN-650 blocks handle specific backward compatible CHI-B, CHI-C, and CHI-D features.

Table 4-46: All blocks backward compatibility

All blocks	CHI-B	CHI-C	CHI-D
DBID[9:8]	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.

Note

If RN-F **TxnID[9:8]** != `0b00`, DCT from a CHI-B or a CHI-C RN-F to a CHI-D RN-F cannot be done.

The following table shows how CMN-650 HN-Fs handle specific backward compatible CHI-B, CHI-C, and CHI-D features.

Table 4-47: HN-F backward compatibility

HN-F protocol	CHI-B	CHI-C	CHI-D
Requests from RN-F	Supported	Supported	Supported
DMT	Yes	Yes	Yes
DCT	Yes	Yes	Yes
Separate response and data	No	Yes	Yes
New fields	MXP drives fixed values.	MXP drives fixed values.	MXP propagates new fields.

The following table shows how CMN-650 DNs handle specific backward compatible CHI-B, CHI-C, and CHI-D features.

Table 4-48: DN backward compatibility

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

The following table shows how CMN-650 CXRAs handle specific backward compatible CHI-B, CHI-C, and CHI-D features.

Table 4-49: CXRA backward compatibility

CXRA protocol	CHI-B	CHI-C	CHI-D
Requests from RN-F	Supported	Supported	Supported
Separate response and data	No	No	No
New fields	MXP drives fixed values.	MXP drives fixed values.	MXP propagates new fields and drives fixed values.

4.15 PCIe integration

CMN-650 supports integration of a PCIe *Root Complex* (RC) or *EndPoint* (EP).

4.15.1 PCIe topology requirements

There are specific topology rules that you must follow when integrating PCIe components with CMN-650.

The following PCIe topology requirements apply:

- PCIe slaves must not be connected to HN-D
- PCIe slaves must not share HN-I with other non-PCIe slaves
- HN-P must only be used to connect to PCIe slaves

4.15.2 PCIe master and slave restrictions and requirements

There are restrictions on both the types and flow of transactions between PCIe devices and CMN-650.

Note

In this section, the term PCIe HN-I refers to an HN-I or HN-P which has a PCIe slave that is connected to it. The term HN-I refers to all other HN-Is.

CMN-650 supports peer-to-peer PCIe traffic. This function allows one PCIe endpoint to communicate with another PCIe endpoint through the interconnect.

Transaction type restrictions

A PCIe master must not send any *Non-Posted Configuration and I/O Writes* (NPR-Wr) targeting CMN-650.

Flow control requirements from CMN-650 to PCIe slave

The PCIe slave must be able to sink at least one NPR-Wr from CMN-650 sent on the PCIe HN-I AXI/ACE-Lite master port. This requirement guarantees that the PCIe HN-I AW channel remains unblocked. Therefore, *Posted Writes* (P-Wrs) from PCIe master targeting the downstream slave device can progress, as required by the PCIe ordering rules.

Flow control requirements from PCIe master to CMN-650

Your configuration might have a *System Memory Management Unit* (SMMU) or GIC-ITS in the path between the PCIe master interface and the RN-I slave interface. If using this configuration, *Non-Posted Reads* (NPR-Rds) from the PCIe master must not target any PCIe HN-I. You can also use one of the following mutually exclusive flow control options:

- Use a separate master 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 masters 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 needs 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

Related information

- 5.3.1.8 por_rnd_s0-2_port_control on page 252
- 5.3.13.8 por_rni_s0-2_port_control on page 639

4.15.3 System requirements for PCIe devices

There are certain system-level requirements that you must meet when integrating PCIe devices with CMN-650. These requirements determine which CMN-650 devices can handle certain request types and how PCIe and non-PCIe transactions must be handled.

Note

In this section, the term PCIe HN-I refers to an HN-I or HN-P which has a PCIe slave connected to it. The term HN-I refers to all other HN-Is.

CMN-650 has the following system requirements for PCIe devices:

- All non-PCIe I/O slave 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 master interface and the RN-I slave 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 5.4.6 RN-I and HN-I PCIe programming sequence on page 844.

4.16 Generic Interrupt Controller communication over AXI4-Stream ports

CMN-650 supports optional master and slave *AXI4-Stream* (A4S) ports on certain blocks for communication between *Generic Interrupt Controller* (GIC) components. A4S ports are supported on RN-I, RN-D, and MXP device ports attached to RN-Fs.

CMN-650 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 LDID, 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 LDIDs:

- The LDID of the target A4S port
- The LDID of the CXRH for GIC traffic targeting the other chip

The discovery process returns the number of A4S ports and LDID 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 4.12 Discovery on page 167.

Other requirements

- The **PUB_DESTID** associated with the GICD A4S port must drive the CXRH **GICD_DESTID** input strap. The **PUB_DESTID** value is included in the CMN ID-mapping file that is created during the IP rendering process. For more information, see the *Arm® Neoverse™ CMN-650 Coherent Mesh Network Configuration and Integration Manual*, which is only available to licensees.
  GICD drives the CMN-650 **RXA4STRI[7:0]** input (8 MSB bits of GICD **ICDRTDEST**), indicating the CCIX link of the target chip. GICD also drives the **RXA4STDEST[7:0]** (8 LSB bits of GICD **ICDRTDEST**) of CXRH for CML SMP configurations.

  Note

  This requirement only applies to two-chip configurations. Contact *Arm* for information on three or more chip configurations.

- The A4S master must assert **valid** irrespective of **ready** state to transmit data.

Related information

- 4.3.2 Logical Device IDs on page 92

4.17 Reliability, Availability, and Serviceability

The CMN-650 *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. The following CMN-650 devices can detect errors:

- XP
- HN-I
- HN-F
- SBSX
- CXHA

The central interrupt handling unit is 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-650 system. For example, the following table shows a possible configuration of the MXP ERRGSRs.

Table 4-50: Example MXP ERRGSR configuration

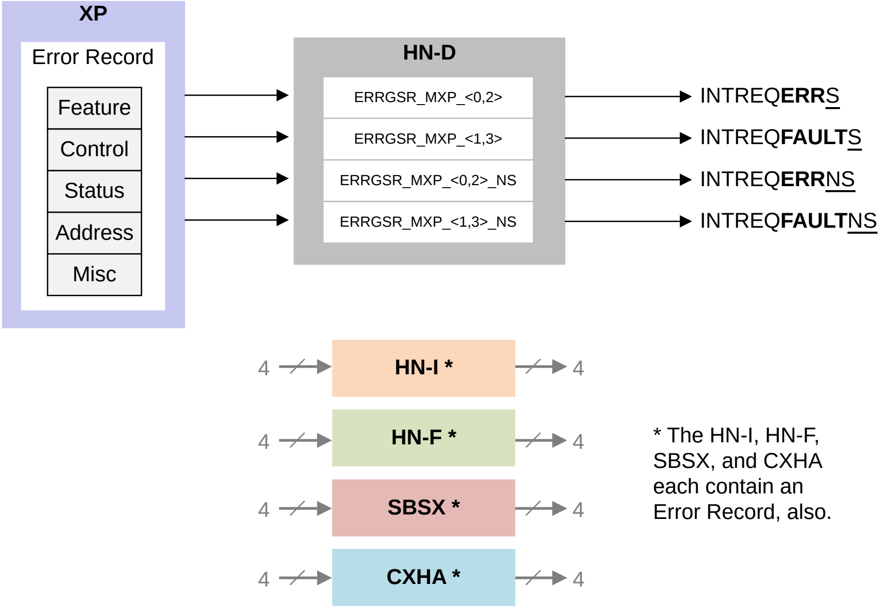
ERRGSR name	Register offset	Error group
por_cfgm_errgsr_mxp_0	16'h3000	MXP<63:0> error status
por_cfgm_errgsr_mxp_1	16'h3008	MXP_<63:0> fault status
por_cfgm_errgsr_mxp_2	16'h3010	MXP_<127:64> error status
por_cfgm_errgsr_mxp_3	16'h3018	MXP_<127:64> fault status
por_cfgm_errgsr_mxp_0_NS	16'h3040	MXP_<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

Note

If CMN-650 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.

The following figure shows the five error groups, and the four respective interrupt request signals, with XP connections highlighted.

Figure 4-52: Error top-level diagram



The HN-I, HN-F, SBSX, and CXHA use the same input/output structure.

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 on error types, see 4.17.1 Error types on page 187.

For register details, see 5.3 Register descriptions on page 243.

Error interrupt handler flow example

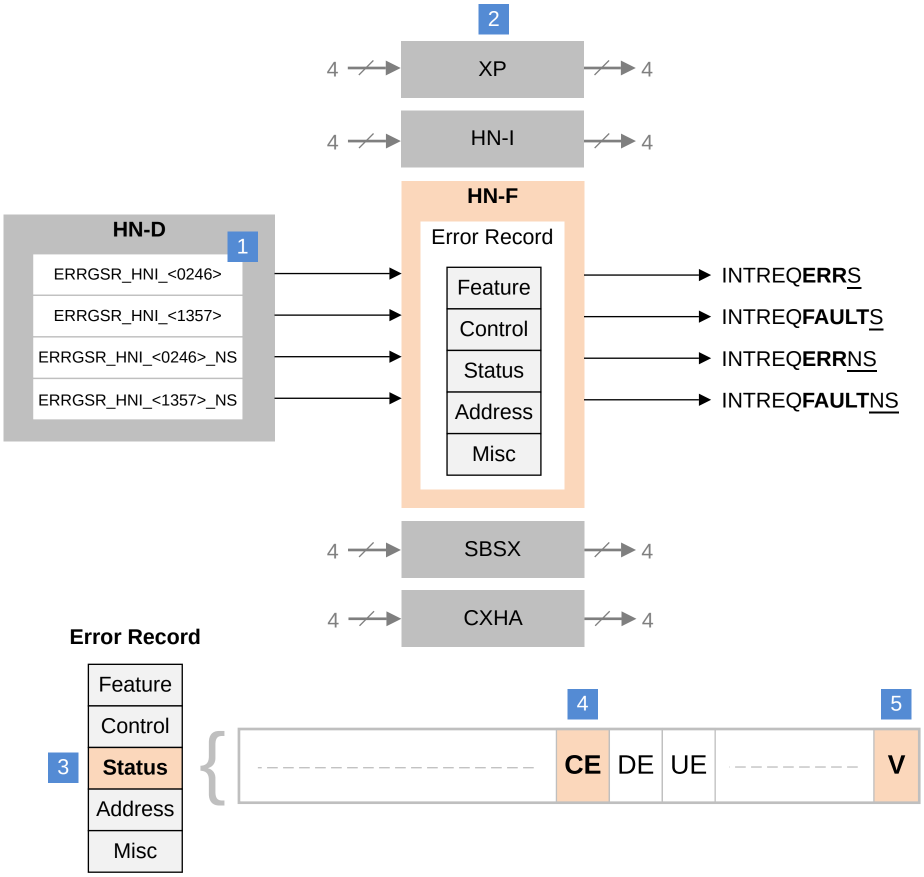
The following sequence of events and figure describe the process for determining the error source and type of an HN-F generating an interrupt request:

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
   - HN-F, which is used in this case
   - SBSX
   - CXHA

   The bit location within the error group indicates the logical ID of that device type. In this case, it reveals an HN-F error, the HN-F Error Record Status block for this example.

3. The Status block of the Error Record for the specific XP, HN-I, HN-F, SBSX, or CXHA indicates the type of error.
4. The Address and Misc blocks of the Error Record provide further details regarding error root cause, in this case a Corrected Error.
5. The Valid bit is also asserted. To clear the asserted interrupt on the pin, the valid bit of the error status has to be cleared.

Figure 4-53: Error interrupt handler flow example



4.17.1 Error types

CMN-650 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, but 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-650.

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-650, 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 through software intervention. They include:

- A data double bit ECC error in the SLC data RAM
- Data check error detected in SLC

The system handles these errors by completing the following steps:

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 by completing the following steps:

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.

CMN-650 follows the *Arm® Reliability, Availability, and Serviceability (RAS) Specification Armv8, for the Armv8-A architecture profile* for mapping of the different error types to the type of interrupt. The following table summarizes the mapping of various error types to the interrupt type.

Table 4-51: Mapping of error types

Interrupt type	Error type		
	Uncorrected error	Detected error	Corrected error
Fault handling	Yes (if ERRCTLR.FI==1)	Yes (if ERRCTLR.FI==1)	Yes (if ERRCTLR.CFI==1)
Error recovery	Yes (if ERRCTLR.UI==1)	No	No

4.17.2 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 4-52: 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

POR_FLIT_PAR_EN	POR_DATACHECK_EN	MXP_DEV_DATACHECK_EN		ED	DE
		P0	P1		
		1	0	2'b01	2'b01
		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.

4.17.3 Error detection, signaling, and reporting

Each CMN-650 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 4-53: 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 4-54: ERRMISC register bit mapping for error reporting units

Bit	Component				
	XP	HN-I	HN-F	SBSX	CXHA
63	-	-	CECOF	-	-
62			SETMATCH		
61			-		

Bit	Component				
	XP	HN-I	HN-F	SBSX	CXHA
60			ERRSET[12:0]		
59					
58	TGTID[10:0]				
57					
56		LPID[4:0]			
55					ERRSET[7:0]
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]		SIZE[2:0]	
29					
28					
27		MEMATTR[3:0]		MEMATTR[3:0]	
26					
25					
24					
23		-		-	

Bit	Component				
	XP	HN-I	HN-F	SBSX	CXHA
22					
21	OPCODE[5:0]	OPCODE[5:0]			
20					
19					
18					
17			OPTYPE[1:0]		
16				OPTYPE	
15	-	-	-	-	
14	SRCID[10:0]	SRCID[10:0]	SRCID[10:0]	SRCID[10:0]	
13					
12					
11					
10					
9					SRCRAID[5:0]
8					
7					
6					
5					
4					
3	-	ERRSRC[3:0]	ERRSRC[3:0]	-	-
2	ERRSRC[2:0]				
1					ERRSRC[1:0]
0					

Error log clearing

In addition to the Error Syndrome Registers, each component has a write-only Error Syndrome Clear Register. Write the applicable mask bits to clear the first_err_vld and mult_err bits of the Error Syndrome 0 Register.

4.17.4 Error reporting rules

CMN-650 uses specific error reporting rules, concerning which errors must be reported and propagated.

The rules regarding error reporting in CMN-650 are:

- Any error originating in CMN-650 is reported
- Any error originating outside CMN-650 but corrupting CMN-650 is reported
- If a response packet from outside CMN-650 does not propagate the response any further, the HN-I can report an error in the response packet

- All non-posted write errors are propagated where possible

4.17.5 HN-F error handling

Errors are reported at the HN-F for various reasons.

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.

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 the HN-F allocates data, then it 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 por_hnf_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.

4.17.6 HN-I error handling

Errors are reported at the HN-I for various reasons.

4.17.6.1 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 or 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 4-55: 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

Request type	Reqerr_cohreq_en controls sending of NDE and log error
Illegal configuration read or write	No

- Coherent reads are downgraded to ReadNoSnp and sent downstream, AXI or ACE-Lite slave
- Coherent and Copyback writes are downgraded to WriteNoSnp and sent downstream, AXI or ACE-Lite slave
- Illegal configuration read is sent as ReadNoSnp to downstream AXI or ACE-Lite slave
- CleanUnique, MakeUnique, atomics, and illegal configuration writes are handled within HN-I
- StashOnceShared, StashOnceUnique, and PrefetchTgt are completed within HN-I without any errors

4.17.6.2 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.

4.17.6.3 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 or ACE-Lite write requests with early completions from HN-I and no request error, HN-I detects the error when it receives the following error response types on the downstream write response (**BRESP**):
  - *Slave Error* (SLVERR)
  - *Decode Error* (DECERR)

  HN-I 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 or 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 or ACE-Lite read requests, SLVERR and poison (if supported by downstream) are both converted to poison within the CMN-650 system. This conversion occurs independent of error on request. DECERRs on downstream read responses are passed on to the requesting RN.

4.17.6.4 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 or write, HN-D only
  - Unsupported exclusive access, HN-P only

Note

For the legal format of configuration read/write request, refer to 5.1.5 Requirements of configuration register reads and writes on page 225.

- Write data error for configuration write request, HN-D only:
  - Partial ByteEnable error
  - Data check error
  - Poison
- AXI or 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 or ACE-Lite response error:
  - SLVERR on **BRESP** for writes with downstream completions

4.17.6.5 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 or 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 or write request, see 5.1.5 Requirements of configuration register reads and writes on page 225.

- 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 or ACE-Lite write requests:
    - Poison error on data if downstream does not support poison
- AXI or ACE-Lite write response error:
    - SLVERR or DECERR on **BRESP** for writes that were sent early completions

4.17.6.6 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. This action is required in CML mode so that NDE error responses are not missed on CCIX because of early completion responses from the CXG block.

4.17.7 SBSX error handling

This section describes how errors are handled at the SBSX.

If the following circumstances are both true, then the SBSX detects and logs errors:

- The AXI memory controller downstream of SBSX does not support POISON, indicated by por_sbsx_unit_info.axdata_poison_en = 0
- CHI write data has poison set

Note

SBSX does not have opcode-based request and response error class as does HN-I.

The following table shows the SBSX summary on sending an NDE or DE.

Table 4-56: SBSX summary on sending NDE or 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	Slave 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	Slave error on **BRESP** from AXI side	DE on COMP for CMOs or writes with EWA = 0

4.17.8 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**.

4.17.9 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 data check. Data check support 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.

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 4-57: XP configuration register flit fields

Error source	Errstatus					Errmisc			
-	DE	CE	MV	UE	V	ERRSRC	SRCID	OPCODE	TGTID
Data Parity P0 REQ channel	1	0	1	0	1	5'b00000	v	v	v
Data Parity P1 REQ channel	1	0	1	0	1	5'b00001	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 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 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
FLIT Parity P0 REQ channel	0	0	1	1	1	5'b00000	v	v	v
FLIT Parity P1 REQ channel	0	0	1	1	1	5'b00001	v	v	v
FLIT Parity P0 RSP channel	0	0	1	1	1	5'b01000	v	v	v
FLIT Parity P1 REQ channel	0	0	1	1	1	5'b01001	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 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

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.

4.17.10 CXHA error handling

Errors are reported at the CXHA for various reasons.

CXHA uses RAMs as buffers for storing the read and write data. The contents of the RAM are protected using byte parity. CXHA reports errors if there is an error that is detected when the contents of the data RAMs are read. These detected errors are of two types:

- Parity error on *Byte-Enable* (BE) fields of the write data RAM
- Parity error on Data, Poison, and Metadata, if enabled, fields of the read and write data RAM

Parity error on BE fields of the write data RAM

The write data buffer RAM stores BE. Parity errors that are detected on BE are treated as UEs. On detecting an error, CXHA does the following:

- Logs the error as UE

Parity error on Data and Poison fields of the read and write data RAM

The read and write data buffer RAMs contain the Data, Poison, and Metadata, if enabled, fields. Errors that are detected on these fields are treated as DEs. If an error is detected on Data fields, then the CXHA does the following:

- Logs the error as DE
- Poisons the data by setting the corresponding poison bit of the data. For more information about data poisoning, see the *AMBA® 5 CHI Architecture Specification*.

If an error is detected on Poison or Metadata, if enabled, fields, then the CXHA does the following:

- Logs the error as DE
- All Poison bits are set to 1

4.17.11 CCIX Protocol Error messaging support

CMN-650 CML supports sending of CCIX *Protocol Error* (PER) message to the CCIX Error Agent present on the Host chip.

CMN-650 includes configuration registers, present in CXLA, and a mechanism to trigger a CCIX PER message. It is expected that an external Error Aggregator/Handler present outside CMN-650 collects and consolidates all the errors and uses these registers to trigger a CCIX PER message to the CCIX Error Agent.

CXLA configuration registers:

- CCIX PER Message Payload:
    - por_cxla_permsg_pyld_0_63
    - por_cxla_permsg_pyld_64_127
    - por_cxla_permsg_pyld_128_191
    - por_cxla_permsg_pyld_192_255
- CCIX PER Message Control:
    - por_cxla_permsg_ctl
- CCIX Error Agent ID:
    - por_cxla_err_agent_id

Mechanism:

- Error Aggregator external to CMN-650:
    - Writes the PER payload in CCIX PER Message Payload registers, por_cxla_permsg_pyld_*
    - Sets per_msg_vld_set bit in CCIX PER Message Control register, por_cxla_permsg_ctl. When set, a PER message is sent on the given CCIX link that the Target ID determines.

  It is the responsibility of CCIX discovery software to program CCIX Error Agent ID in CCIX Error Agent ID por_cxla_err_agent_id register. This programming should happen during initial system bring up and the programmed ID is used as the target ID on CCIX PER message.

  PER message is supported only in non-SMP mode. For more details on SMP mode, see 4.11.7 CML Symmetric Multiprocessor support on page 163.

  By default, *Error SourceID* (ESID) field from PER message payload, bits [53:48], are used as source ID on PER message. per_msg_srcid_ovrd and per_msg_srcid fields in CCIX PER message control register, por_cxla_permsg_ctl, can be used to override source ID sent on PER message.

Note

CMN-650 CML does not implement a CCIX Error Agent. It can accept the incoming PER messages, but these messages are dropped at CXLA.

4.18 Transaction handling

The handling of certain CHI transaction types and fields differs according to the CMN-650 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.

4.18.1 Atomics

CMN-650 supports atomic accesses to both cacheable and non-cacheable memory locations.

4.18.1.1 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 process the transaction in the following way:

1. Issue a read to the SN
2. Atomically update the copy of the data in the HN-F
3. 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.

4.18.1.2 Atomic requests in SN

The SN node (CHI memory controller or SBSX bridge) does not process atomic requests.

4.18.1.3 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 master.

4.18.1.4 Atomic requests in RN-I and RN-D

RN-I and RN-D support atomic transactions in CMN-650. These nodes can receive atomics from ACE5-Lite and AXI5 masters, and translate them on CHI before sending them to HN-F, HN-I, or CXRH 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. The *NUM_ATOMIC_BUF* parameter determines the depth of this buffer.

Note

For atomic transactions arriving at RN-I and RN-D from ACE5-Lite and AXI5 masters, all write strobes within **AWSIZE** must be set. RN-I and RN-D do not allow sparse write strobes for atomic transactions.

4.18.2 Exclusive accesses

CMN-650 supports exclusive accesses to both Shareable and Non-shareable locations.

For more information, see the *AMBA® 5 CHI Architecture Specification*.

4.18.2.1 Exclusive accesses in HN-F

The HN-F supports exclusive access on ReadNoSnp, 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. ReadShared, ReadClean, ReadNotSharedDirty, and CleanUnique exclusives are used for shareable and coherent memory locations. Each HN-F partition includes 64 exclusive monitors for tracking of these transaction types. Each monitor can act as both a PoC monitor and System monitor, as the *AMBA® 5 CHI Architecture Specification* defines.

Only 64 unique logical threads, which are designated by a unique combination of SrcID and LPID, can concurrently access the HN-F exclusive monitors.

4.18.2.2 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.

Note

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.

4.18.2.3 CML support for exclusive accesses

CMN-650 CML supports exclusive accesses in some circumstances. Support for these transactions and the guidance for configuration depends on whether SMP mode is enabled.

SMP mode

In SMP mode, CMN-650 CML supports remote exclusive accesses from RN-Fs.

Remote exclusive accesses from an RN-I or RN-D are not supported.

Support for remote exclusive accesses in *CCIX Gateway* (CXG) blocks include these constraints:

- CXRA in local CXG block passes Excl and LPID fields of incoming CHI request on CCIX request message USER (Ext) field.
- CXHA in the remote CCIX gateway (CXG) block extracts these bits from CCIX request message USER (Ext) field. CXHA then sends these bits on respective CHI Excl and LPID fields. CXHA sets the source type as RN-F based on its RAID to LDID register.
- Exclusive OK (EXOK) response is sent as `0b01` on CCIX RespErr field.

Note

`0b01` is a reserved encoding in RespErr field and this field is sent as a CCIX extension (Ext6).

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.

Non-SMP mode

In non-SMP mode, we recommend setting the lnk<X>_excl_load_dwngrd and lnk<X>_excl_store_dwngrd bits in the respective por_cxg_ra_cxprtcl_link<X>_ctl register. If these bits are set, a shareable exclusive access, for example a load or store access, is downgraded to a shareable non-exclusive access. In other words, the Excl attribute is dropped from the access. The access is then sent on the corresponding CCIX link. Any incoming OK response is converted to EXOK when sent to the requesting CHI RN.

In non-SMP mode, exclusive accesses are not supported to Normal Non-cacheable or Device memory. Load exclusives are sent as Normal loads and the incoming OK response is passed to the requesting RN. Store exclusives are terminated at CXRA with an NDE response.

CXRA also has more bits to override the RespErr field on response for an exclusive access in non-SMP mode. The lnk<X>_excl_resperr_value value in the por_cxg_ra_cxprtcl_link<X>_ctl register can override incoming OK responses. You must take care when using these bits, because response overrides are not expected for normal use.

4.18.2.4 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 provide 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.

4.18.3 Barriers

Barriers were deprecated from CHI-B onwards. All masters (fully coherent and I/O coherent) must handle barriers at the source.

When memory barrier ordering or completion guarantees are required, masters 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 might be initiated by an Arm DSB instruction, 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*.

4.18.4 Distributed Virtual Memory messages

If an RN-F supports *Distributed Virtual Memory* (DVM) messages, it can send DVM requests and receive DVM snoops.

A DVM message from an RN-F is sent to the HN-D. On receiving the DVM message, the HN-D:

- Forwards the DVM message as a snoop to the participating RNs and CXRAs
- Collects the individual snoop responses
- Sends a single response back to the RN-F that originated the DVM message transaction

The **SYSCOREQ**/**SYSCOACK** mechanism provides proxy snoop responses in scenarios when the RN is powered down. For more **SYSCOREQ**/**SYSCOACK** information, refer to 4.2.9 RN entry to and exit from snoop and DVM domains on page 85.

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

4.18.4.1 Support for early completion of DVMOp requests

CMN-650 HN-D and CXRA 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:

HN-D disable_dvmop_early_comp bit in por_dn_aux_ctl register

CXRA dvm_earlycomp_en bit in por_cxg_ra_aux_ctl register

By default, the early DVMOp completion mode is disabled at HN-D. 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

CXRA can give early completions for DVMOp requests that are sent over a CCIX SMP link. By default, this mode is enabled at CXRA. When this mode is enabled, any NDE on DVMOp completion is dropped.

4.18.5 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-650 implements the CBusy indication function in the following node types:

- HN-F
- SBSX
- CXRA

HN-I, HN-T, HN-D, and HN-P always drive the CBusy values as `0b000`.

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 4-58: HN-F POCQ CBusy thresholds

CBusy[2:0]	Tracker occupancy level
`0b011`	≥24
`0b010`	≥16
`0b001`	≥8
`0b000`	<8

HN-F also supports a CBusy multi-source mode. The CBusy[2] bit indicates that multiple sources, or RNs, have outstanding requests pending in the HN-F POCQ. HN-F also supports a mode where, when calculating CBusy[2], it excludes outstanding RN-I requests in the POCQ. This mode is enabled if the hnf_cbusy_mtbit_exclude_rni field of the por_hnf_cbusy_limit_ctl register is set to `0b1`. If this field is set, then HN-F ignores outstanding requests from the RN-I while calculating the CBusy[2] value.

SBSX CBusy

SBSXs only drive CBusy on returning TXDAT flits targeting RNs. The SBSX uses 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 4-59: 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`.

CXRA CBusy

CXRA 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 CXRA outgoing CBusy in all responses to RNs.

Table 4-60: CXRA RHT CBusy thresholds

CBusy[2:0]	Tracker occupancy level
`0b011`	≥192
`0b010`	≥128
`0b001`	≥64
`0b000`	<64

CXRAs do not use the multi-source mode bit, so CBusy[2] is always set to `0b0`.

4.18.5.1 Advanced CBusy handling in HN-F

CMN-650 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.
- 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.

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*, 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 por_hnf_cbusy_limit_ctl_register. This register controls the HN-F CBusy threshold for Read requests.

Table 4-61: por_hnf_cbusy_limit_ctl register for CBusy thresholds (all requests or read types)

Bits	Name	Description
[63]	hnf_cbusy_mtbit_exclude_rni	Allows HN-F to ignore outstanding read requests from RN-I when calculating busyness.
[48]	hnf_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.
[23:16]	hnf_cbusy_high_limit	Specifies the POC valid threshold at which HN-F is considered very busy.
[15:8]	hnf_cbusy_med_limit	Specifies the POC valid threshold at which HN-F is considered medium busy.
[7:0]	hnf_cbusy_low_limit	Specifies the POC valid threshold at which HN-F is considered least busy.

The following table shows the format of the por_hnf_cbusy_write_limit_ctl register. This register controls the HN-F CBusy threshold for write requests.

Table 4-62: Register for CBusy thresholds (write requests)

Bit field	Field	Description
[48]	hnf_cbusy_sep_copyback_types	When set, HN-F calculates CBusy based on outstanding CopyBack and NonCopyBack type requests independently in the HN-F POCQ.
[23:16]	hnf_cbusy_high_limit	Specifies the POC valid threshold at which HN-F is considered very busy.
[15:8]	hnf_cbusy_med_limit	Specifies the POC valid threshold at HN-F is considered medium busy.

Bit field	Field	Description
[7:0]	hnf_cbusy_low_limit	Specifies the POC valid threshold at which HN-F is considered least busy.

The following table shows the CBusy values that are returned to RNs according to programming.

Table 4-63: HN-F CBusy value propagation according to programming

hnf_adv_cbusy_mode_en	hnf_cbusy_rd_wr_types_en	sn_cbusy_prop_en	cbusy_highest_of_all_en	CBusy value passed to RN
0b0	x	x	x	POCQ CBusy value is returned.
0b1	0b0	x	x	POCQ CBusy value is returned.
0b1	0b1	0b0	0b0	POCQ CBusy value for read or write is returned, according to the request type.
0b1	0b1	0b1	0b0	SN CBusy value for read or write is returned for the corresponding SN group, according to the request type.
0b1	0b1	x	0b1	Highest of either the SN or POCQ CBusy value for read or write is returned, according to the request type.

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 hnf_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*). 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, if HN-F receives ≥16 transactions and CBusy = 0b11 from Group 0 SN-Fs, then HN-F treats the SN-F CBusy value as 0b11. This value corresponds to very busy.

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 fixed number of transactions outstanding at any point for a given SN group and request type:
    - CBusy = `0b11` (very busy): HN-F restricts transactions to a maximum of a quarter of the number of POCQ entries.
    - CBusy = `0b10` (medium busy): HN-F restricts transactions to a maximum of half of the number of POCQ entries.
    - CBusy = `0b01` (low busy): HN-F restricts transactions to three quarters of the number of POCQ entries.
    - 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 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.

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 por_hnf_cbusy_resp_ctl register. This register controls the CBusy responses.

Table 4-64: por_hnf_cbusy_resp_ctl register for configuring CBusy value on responses

Bits	Name	Description: Controls the CBusy responses
[21:16]	cbusy_sn_dynamic_ot_count	Count by which the OT count is incremented or decremented for dynamic OT throttling.
[7]	cbusy_sn_static_ot_mode_en	Enables the static OT throttling to SN.
[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.
[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 por_hnf_cbusy_limit_ctl and por_hnf_cbusy_write_limit_ctl.

The following table shows the format of the por_hnf_sam_sn_properties1 register. This register controls the group to which each SN belongs.

Table 4-65: Register for identifying SN groups

Bit field	Field	Description

[57]	Region1_sn_group	`0b0`
		Group A
		`0b1`
		Group B

[49]	Region0_sn_group	
[41]	sn5_group	
[33]	sn4_group	
[25]	sn3_group	
[17]	sn2_group	
[9]	sn1_group	
[1]	sn0_group	

The following table shows the format of the por_hnf_cbusy_sn_ctl register. This register controls the CBusy sampling.

Table 4-66: por_hnf_cbusy_sn_ctl register for CBusy sampling control

Bit field	Field	Description
[56:48]	hnf_cbusy_txn_cnt	Number of SN responses over which the CBusy counters are tracked.
[41:32]	hnf_cbusy_threshold_cntr11	CBusy threshold at which SN-F is considered busy for Counter_11.
[25:16]	hnf_cbusy_threshold_cntr10	CBusy threshold at which SN-F is considered busy for Counter_10.
[9:0]	hnf_cbusy_threshold_cntr01	CBusy threshold at which SN-F is considered busy for Counter_01.

HN-F continues to propagate the multi-source bit (CBusy[2]) in the advanced modes.

4.18.6 REQ RSVDC propagation

CMN-650 supports propagation of the RSVDC field of the CHI REQ flit through the interconnect.

For a multi-chip system, the REQ RSVDC field is preserved and only passed over a CCIX SMP link. For AXI slave interfaces, the incoming **AxUSER** field is mapped to CHI REQ RSVDC field and is propagated through the interconnect. For AXI master 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.

4.18.7 DAT RSVDC propagation

CMN-650 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 CCIX traffic when in SMP mode.

This feature is enabled by setting the *META_DATA_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 CXRA, CXHA, or CXSA traffic.

The **RUSER** and **WUSER** signal widths increase on ACE-Lite master and slave interfaces when *META_DATA_EN* is set to 1. However they are not used to propagate DAT RSVDC values.

Note

Partial cache states are not supported when the *META_DATA_EN* parameter is set.

4.19 Processor events

CMN-650 supports communicating processor events to all processors in the system.

Refer to the processor event interface signals described in B.14 Processor event interface signals on page 988.

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.

4.20 Quality of Service

CMN-650 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 response latency that the transactions of devices incur has a high impact on the performance of these devices. 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-650 microarchitecture. Each component in CMN-650 contributes to the overall QoS microarchitecture.

4.20.1 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-650 components use the QPV to provide prioritized arbitration and to prevent Head-of-line blocking based on the QPV.

4.20.2 Microarchitectural QoS support

The QPV of RN requests must be modulated depending on how well or poorly their respective QoS requirements are met.

4.20.2.1 QoS regulators

CMN-650 supports end-to-end *Quality-of-Service* (QoS) which guarantees using QoS mechanisms that are distributed throughout the system. QoS-modulation capability can be integrated into the RN. However CMN-650 enables system designers to include non-QoS-aware devices in the CMN-650 system, but still have these devices meet the QoS-modulation requirements of the CMN-650 QoS microarchitecture.

The QoS provision uses the QoS field in each RN request to influence arbitration priority at every QoS decision point. The QoS field is then propagated through all subsequent packets that are generated by the transaction. 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-650.

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-650 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 QPV field. The QR adjusts the QPV field upwards for higher priority in the system and downwards for lower priority.

QoS regulators are present at each RN-I ACE-Lite interface and at each XP device port that is connected to an RN-F.

CMN-650 QoS regulators have three operating modes:

- Pass-through
- Programmed QoS value
- Regulation

These operating modes are controlled through memory-mapped configuration registers.

4.20.2.2 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 the same 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.

4.20.2.3 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.

4.20.2.4 HN-F QoS support

The HN-F is a key shared system resource that is used for system caching and for communication with the memory controller for external memory access.

The HN-F includes the following QoS support mechanisms:

QoS decoding in HN-F

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

Table 4-67: QoS classes in HN-F

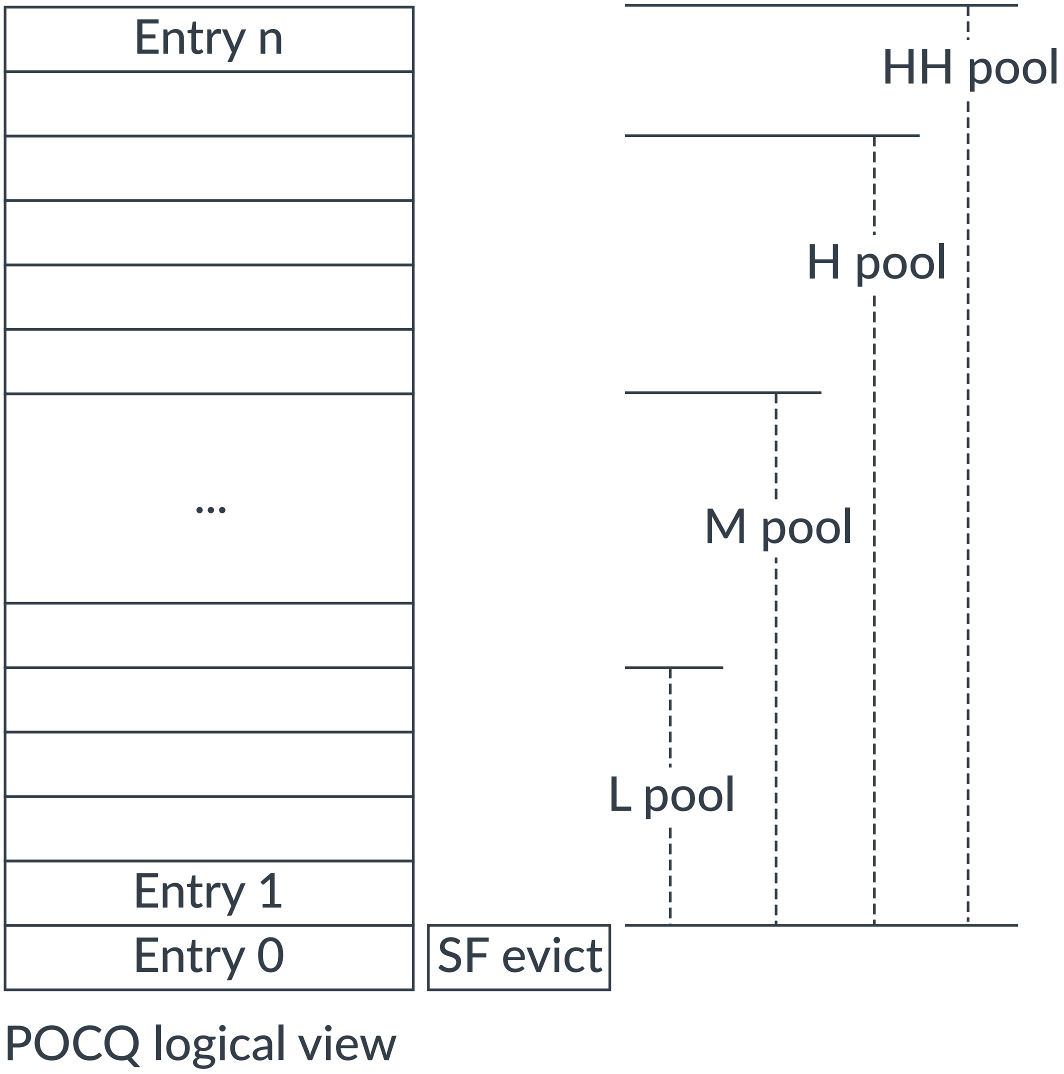
QoS value range	QoS class	Class mnemonic	Priority
15	HighHigh	HH	Highest
14-12	High	H	High
11-8	Med	M	Medium
7-0	Low	L	Low

QoS class and POCQ resource availability

The HN-F includes a 32-entry or 64-entry structure, the POCQ, from which all transaction ordering and scheduling is performed. The POCQ buffers are shared resources for all QoS classes, with one entry being reserved for internal use. The higher the QoS class, the higher the occupancy availability. As the following figure shows, the POCQ is partitioned so that higher priority requests are able to use a larger percentage of the POCQ buffering. This partitioning ensures bandwidth and latency requirements of higher priority transactions are met.

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

Figure 4-54: POCQ availability and QoS classes



The QoS pools are:

hh_pool	Available for HH class.
h_pool	Available for H class and HH class.
m_pool	Available for M class, H class, and HH class.
l_pool	Available for all classes.
seq	SF evictions only.

Warning

Register fields must be programmed so that highhigh_qos_max_cnt > high_qos_max_cnt > med_qos_max_cnt > low_qos_max_cnt ≥ 2. The maximum value that is allowed for highhigh_qos_max_cnt = (*NUM_ENTRIES_POCQ* - seq_qos_max_cnt).

POCQ allocation policies

Allocation into POCQ entries can follow either of two paths:

- Immediate allocation on receipt of the initial request by the HN-F
- Allocation of a retried request after a protocol-layer retry of the initial request

The first case is the expected common case and is always the case in a reasonably uncongested system. If the POCQ has an available buffer corresponding to the QoS class of an arriving request, that request allocates in the POCQ.

However, consider a congested system and a POCQ entry with the QoS class of an arriving request is unavailable at the time of arrival. In this case, the AMBA® 5 CHI protocol requires that the arriving request receives a protocol-layer retry. The transaction flow in this case is as follows:

1. A request arrives at a congested HN-F.
2. The HN-F does not have an available POCQ buffer corresponding to the QoS class of the new transaction.
3. The HN-F increments a credit counter for the specific QoS class of the specific RN and sends a RetryAck response to the RN.
4. On receiving a RetryAck response, the RN then waits for a follow-on PCreditGrant response.
5. When a POCQ entry becomes available, the HN-F reserves that buffer for the highest-priority RN with a nonzero credit-counter. It also sends a PCreditGrant response to the selected RN.
6. On receiving a PCreditGrant, the RN re-issues the transaction, which is guaranteed to be allocated into the HN-F.

This mechanism serves as prioritized arbitration based on QoS values for requests that are sent to the HN-F.

POCQ scheduler policies

When transactions are allocated into the POCQ, they are scheduled for issuance based on the QPV as follows, in descending order of priority:

- Starved transactions. These transactions are lower-priority transactions that have made no progress for a specific number of cycles. The number of cycles is specified in the respective fields in the RN Starvation Register.
- Highest QoS class
- Round-robin arbitration within a QoS class among the issuable transactions

4.20.2.5 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 it is full, requests are 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. The scheduling is 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.

4.20.2.6 CML QoS overrides

You can program CMN-650 to override the QPV on incoming transactions through the HA.

You can program the HA config control register, por_cxg_ha_cfg_ctl, 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.

4.20.3 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-650 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, refer to Table 4-67: QoS classes in HN-F on page 217.

QoS regulator settings

To meet the bandwidth and latency requirements of the described system configuration, CMN-650 QoS regulators can be configured with the settings as described in the following table.

Table 4-68: 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. However, the programmed latency must not violate the maximum latency requirement.

To meet the bandwidth and latency requirements of the described system configuration, HN-F QoS reservation values can be configured as summarized in the following table. The table is based on 32-entry POCQ with one entry for SF back invalidations.

Table 4-69: 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. Therefore the processor QPV generally achieves minimum latency, except when there is high-bandwidth real-time traffic.
- Real-time devices 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-650 preventing Head-of-line blocking from lower-priority or higher-latency transactions.

5 Programmers model

This chapter describes the application-level registers and provides an overview for programming the CMN-650 interconnect.

5.1 About the programmers model

A CMN-650 interconnect consists of various components, such as XPs, RN-Is, or DTCs, 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-650 HN-D APB interface.

A full description of a CMN-650 interconnect consists of:

- A list of components
- The compile-time configuration options for each component
- The connectivity between the components

Software can determine the full configuration of the CMN-650 interconnect through a sequence of accesses to the configuration register space.

5.1.1 Node configuration register address mapping

All CMN-650 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 and 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-650 interconnect are grouped into 64KB regions each associated with a CMN-650 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-650 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 on these steps, see 4.12.4 Discovery tree structure on page 175.

With this sequence, software can build a list of all components in the system and the addresses of their respective 64KB configuration regions.

5.1.2 Global configuration register region

The 64KB block at offset `0x0` contains global information and configuration for CMN-650, 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 5.3.9 Configuration master register descriptions on page 507.

5.1.3 XP configuration register region

Each XP has a 64KB configuration register region with information about that XP and all associated components.

Refer to 5.3.14 XP register descriptions on page 649 for more information.

5.1.4 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 5-1: 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

Register sections	Relative offset	Absolute offset	Description
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
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

5.1.5 Requirements of configuration register reads and writes

Reads and writes to the CMN-650 configuration registers must meet certain requirements.

A dedicated APB slave port is provided for the access of all CMN-650 configuration registers. The APB slave 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.

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 byte 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.
- Secure registers can only be accessed through a Secure access, that is, NS = 0. Non-secure registers can be accessed through either a Secure or Non-secure access.

For more information on error signal handling, see 4.17 Reliability, Availability, and Serviceability on page 184.

5.2 Register summary

The register summary tables list the registers in CMN-650.

5.2.1 RN-D register summary

This section lists the RN-D registers used in CMN-650.

Table 5-2: por_rnd_cfg register summary

Offset	Name	Type	Description
16'h0	por_rnd_node_info	RO	5.3.1.1 por_rnd_node_info on page 243
16'h80	por_rnd_child_info	RO	5.3.1.2 por_rnd_child_info on page 244
16'h980	por_rnd_secure_register_groups_override	RW	5.3.1.3 por_rnd_secure_register_groups_override on page 245
16'h900	por_rnd_unit_info	RO	5.3.1.4 por_rnd_unit_info on page 246
16'h908	por_rnd_unit_info2	RO	5.3.1.5 por_rnd_unit_info2 on page 247
16'hA00	por_rnd_cfg_ctl	RW	5.3.1.6 por_rnd_cfg_ctl on page 248
16'hA08	por_rnd_aux_ctl	RW	5.3.1.7 por_rnd_aux_ctl on page 250
16'hA10 + (8 × #[0, 1, 2])	por_rnd_s0-2_port_control	RW	5.3.1.8 por_rnd_s0-2_port_control on page 252
16'hA28 + (8 × #[0, 1, 2])	por_rnd_s0-2_mpam_control	RW	5.3.1.9 por_rnd_s0-2_mpam_control on page 253
16'hA80 + (32 × #[0, 1, 2])	por_rnd_s0-2_qos_control	RW	5.3.1.10 por_rnd_s0-2_qos_control on page 254
16'hA88 + (32 × #[0, 1, 2])	por_rnd_s0-2_qos_lat_tgt	RW	5.3.1.11 por_rnd_s0-2_qos_lat_tgt on page 256
16'hA90 + (32 × #[0, 1, 2])	por_rnd_s0-2_qos_lat_scale	RW	5.3.1.12 por_rnd_s0-2_qos_lat_scale on page 257
16'hA98 + (32 × #[0, 1, 2])	por_rnd_s0-2_qos_lat_range	RW	5.3.1.13 por_rnd_s0-2_qos_lat_range on page 259
16'h2000	por_rnd_pmu_event_sel	RW	5.3.1.14 por_rnd_pmu_event_sel on page 260
16'h1C00	por_rnd_syscoreq_ctl	RW	5.3.1.15 por_rnd_syscoreq_ctl on page 262
16'h1C08	por_rnd_syscoack_status	RO	5.3.1.16 por_rnd_syscoack_status on page 263

5.2.2 CXRA register summary

This section lists the CXRA registers used in CMN-650.

Table 5-3: por_cxg_ra_cfg register summary

Offset	Name	Type	Description
16'h0	por_cxg_ra_node_info	RO	5.3.2.1 por_cxg_ra_node_info on page 265
16'h80	por_cxg_ra_child_info	RO	5.3.2.2 por_cxg_ra_child_info on page 265
16'h980	por_cxg_ra_secure_register_groups_override	RW	5.3.2.3 por_cxg_ra_secure_register_groups_override on page 266
16'h900	por_cxg_ra_unit_info	RO	5.3.2.4 por_cxg_ra_unit_info on page 267
16'hA00	por_cxg_ra_cfg_ctl	RW	5.3.2.5 por_cxg_ra_cfg_ctl on page 268
16'hA08	por_cxg_ra_aux_ctl	RW	5.3.2.6 por_cxg_ra_aux_ctl on page 270
16'hA18	por_cxg_ra_cbusy_limit_ctl	RW	5.3.2.7 por_cxg_ra_cbusy_limit_ctl on page 271
16'hC00 + (8 × #[0, 1, ... 7])	por_cxg_ra_sam_addr_region_reg0-7	RW	5.3.2.8 por_cxg_ra_sam_addr_region_reg0-7 on page 272
16'hD00	por_cxg_ra_agentid_to_linkid_val	RW	5.3.2.9 por_cxg_ra_agentid_to_linkid_val on page 273
16'hD10 + (8 × #[0, 1, ... 7])	por_cxg_ra_agentid_to_linkid_reg0-7	RW	5.3.2.10 por_cxg_ra_agentid_to_linkid_reg0-7 on page 274
16'hE00 + (8 × #[0, 1, ... 9])	por_cxg_ra_rni_ldid_to_exp_raid_reg0-9	RW	5.3.2.11 por_cxg_ra_rni_ldid_to_exp_raid_reg0-9 on page 276
16'hF00 + (8 × #[0, 1, ... 9])	por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9	RW	5.3.2.12 por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9 on page 277
16'h1000 + (8 × #[0, 1, ... 127])	por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127	RW	5.3.2.13 por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127 on page 278
16'h1400 + (8 × #[0, 1, ... 127])	por_cxg_ra_rnf_ldid_to_nodeid_reg0-127	RO	5.3.2.14 por_cxg_ra_rnf_ldid_to_nodeid_reg0-127 on page 279
16'h1800 + (8 × #[0, 1, ... 127])	por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127	RW	5.3.2.15 por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127 on page 281
16'h2000	por_cxg_ra_pmu_event_sel	RW	5.3.2.16 por_cxg_ra_pmu_event_sel on page 282
16'h1C00	por_cxg_ra_cxprtcl_link0_ctl	RW	5.3.2.17 por_cxg_ra_cxprtcl_link0_ctl on page 283
16'h1C08	por_cxg_ra_cxprtcl_link0_status	RO	5.3.2.18 por_cxg_ra_cxprtcl_link0_status on page 286
16'h1C10	por_cxg_ra_cxprtcl_link1_ctl	RW	5.3.2.19 por_cxg_ra_cxprtcl_link1_ctl on page 287
16'h1C18	por_cxg_ra_cxprtcl_link1_status	RO	5.3.2.20 por_cxg_ra_cxprtcl_link1_status on page 290
16'h1C20	por_cxg_ra_cxprtcl_link2_ctl	RW	5.3.2.21 por_cxg_ra_cxprtcl_link2_ctl on page 291
16'h1C28	por_cxg_ra_cxprtcl_link2_status	RO	5.3.2.22 por_cxg_ra_cxprtcl_link2_status on page 294

5.2.3 HN-I register summary

This section lists the HN-I registers used in CMN-650.

Table 5-4: por_hni_cfg register summary

Offset	Name	Type	Description
16'h0	por_hni_node_info	RO	5.3.3.1 por_hni_node_info on page 295
16'h80	por_hni_child_info	RO	5.3.3.2 por_hni_child_info on page 296
16'h980	por_hni_secure_register_groups_override	RW	5.3.3.3 por_hni_secure_register_groups_override on page 297
16'h900	por_hni_unit_info	RO	5.3.3.4 por_hni_unit_info on page 298
16'hC00	por_hni_sam_addrregion0_cfg	RW	5.3.3.5 por_hni_sam_addrregion0_cfg on page 300
16'hC08	por_hni_sam_addrregion1_cfg	RW	5.3.3.6 por_hni_sam_addrregion1_cfg on page 301
16'hC10	por_hni_sam_addrregion2_cfg	RW	5.3.3.7 por_hni_sam_addrregion2_cfg on page 302
16'hC18	por_hni_sam_addrregion3_cfg	RW	5.3.3.8 por_hni_sam_addrregion3_cfg on page 304
16'hA00	por_hni_cfg_ctl	RW	5.3.3.9 por_hni_cfg_ctl on page 305
16'hA08	por_hni_aux_ctl	RW	5.3.3.10 por_hni_aux_ctl on page 306
16'h3000	por_hni_errfr	RO	5.3.3.11 por_hni_errfr on page 308
16'h3008	por_hni_errctlr	RW	5.3.3.12 por_hni_errctlr on page 309
16'h3010	por_hni_errstatus	W1C	5.3.3.13 por_hni_errstatus on page 310
16'h3018	por_hni_erraddr	RW	5.3.3.14 por_hni_erraddr on page 311
16'h3020	por_hni_errmisc	RW	5.3.3.15 por_hni_errmisc on page 312
16'h3100	por_hni_errfr_NS	RO	5.3.3.16 por_hni_errfr_NS on page 314
16'h3108	por_hni_errctlr_NS	RW	5.3.3.17 por_hni_errctlr_NS on page 315
16'h3110	por_hni_errstatus_NS	W1C	5.3.3.18 por_hni_errstatus_NS on page 316
16'h3118	por_hni_erraddr_NS	RW	5.3.3.19 por_hni_erraddr_NS on page 318
16'h3120	por_hni_errmisc_NS	RW	5.3.3.20 por_hni_errmisc_NS on page 319
16'h2000	por_hni_pmu_event_sel	RW	5.3.3.21 por_hni_pmu_event_sel on page 320
16'h2008	por_hnp_pmu_event_sel	RW	5.3.3.22 por_hnp_pmu_event_sel on page 322

5.2.4 CXLA register summary

This section lists the CXLA registers used in CMN-650.

Table 5-5: por_cxla_cfg register summary

Offset	Name	Type	Description
16'h0	por_cxla_node_info	RO	5.3.4.1 por_cxla_node_info on page 324
16'h80	por_cxla_child_info	RO	5.3.4.2 por_cxla_child_info on page 325
16'h980	por_cxla_secure_register_groups_override	RW	5.3.4.3 por_cxla_secure_register_groups_override on page 326
16'h900	por_cxla_unit_info	RO	5.3.4.4 por_cxla_unit_info on page 327
16'hA00	por_cxla_cfg_ctl	RW	5.3.4.5 por_cxla_cfg_ctl on page 328
16'hA08	por_cxla_aux_ctl	RW	5.3.4.6 por_cxla_aux_ctl on page 329
16'hC00	por_cxla_ccix_prop_capabilities	RO	5.3.4.7 por_cxla_ccix_prop_capabilities on page 333

Offset	Name	Type	Description
16'hC08	por_cxla_ccix_prop_configured	RW	5.3.4.8 por_cxla_ccix_prop_configured on page 335
16'hC10	por_cxla_tx_cxs_attr_capabilities	RO	5.3.4.9 por_cxla_tx_cxs_attr_capabilities on page 336
16'hC18	por_cxla_rx_cxs_attr_capabilities	RO	5.3.4.10 por_cxla_rx_cxs_attr_capabilities on page 338
16'hC30	por_cxla_agentid_to_linkid_reg0	RW	5.3.4.11 por_cxla_agentid_to_linkid_reg0 on page 340
16'hC38	por_cxla_agentid_to_linkid_reg1	RW	5.3.4.12 por_cxla_agentid_to_linkid_reg1 on page 341
16'hC40	por_cxla_agentid_to_linkid_reg2	RW	5.3.4.13 por_cxla_agentid_to_linkid_reg2 on page 343
16'hC48	por_cxla_agentid_to_linkid_reg3	RW	5.3.4.14 por_cxla_agentid_to_linkid_reg3 on page 344
16'hC50	por_cxla_agentid_to_linkid_reg4	RW	5.3.4.15 por_cxla_agentid_to_linkid_reg4 on page 345
16'hC58	por_cxla_agentid_to_linkid_reg5	RW	5.3.4.16 por_cxla_agentid_to_linkid_reg5 on page 347
16'hC60	por_cxla_agentid_to_linkid_reg6	RW	5.3.4.17 por_cxla_agentid_to_linkid_reg6 on page 348
16'hC68	por_cxla_agentid_to_linkid_reg7	RW	5.3.4.18 por_cxla_agentid_to_linkid_reg7 on page 349
16'hC70	por_cxla_agentid_to_linkid_val	RW	5.3.4.19 por_cxla_agentid_to_linkid_val on page 351
16'hC78	por_cxla_linkid_to_pcie_bus_num	RW	5.3.4.20 por_cxla_linkid_to_pcie_bus_num on page 351
16'hC80	por_cxla_tlp_hdr_fields	RW	5.3.4.21 por_cxla_tlp_hdr_fields on page 352
16'hD00	por_cxla_permsg_pyld_0_63	RW	5.3.4.22 por_cxla_permsg_pyld_0_63 on page 354
16'hD08	por_cxla_permsg_pyld_64_127	RW	5.3.4.23 por_cxla_permsg_pyld_64_127 on page 354
16'hD10	por_cxla_permsg_pyld_128_191	RW	5.3.4.24 por_cxla_permsg_pyld_128_191 on page 355
16'hD18	por_cxla_permsg_pyld_192_255	RW	5.3.4.25 por_cxla_permsg_pyld_192_255 on page 356
16'hD20	por_cxla_permsg_ctl	RW	5.3.4.26 por_cxla_permsg_ctl on page 357
16'hD28	por_cxla_err_agent_id	RW	5.3.4.27 por_cxla_err_agent_id on page 358
16'hD30	por_cxla_agentid_to_portid_reg0	RW	5.3.4.28 por_cxla_agentid_to_portid_reg0 on page 359
16'hD38	por_cxla_agentid_to_portid_reg1	RW	5.3.4.29 por_cxla_agentid_to_portid_reg1 on page 360
16'hD40	por_cxla_agentid_to_portid_reg2	RW	5.3.4.30 por_cxla_agentid_to_portid_reg2 on page 361
16'hD48	por_cxla_agentid_to_portid_reg3	RW	5.3.4.31 por_cxla_agentid_to_portid_reg3 on page 363
16'hD50	por_cxla_agentid_to_portid_reg4	RW	5.3.4.32 por_cxla_agentid_to_portid_reg4 on page 364
16'hD58	por_cxla_agentid_to_portid_reg5	RW	5.3.4.33 por_cxla_agentid_to_portid_reg5 on page 365
16'hD60	por_cxla_agentid_to_portid_reg6	RW	5.3.4.34 por_cxla_agentid_to_portid_reg6 on page 367
16'hD68	por_cxla_agentid_to_portid_reg7	RW	5.3.4.35 por_cxla_agentid_to_portid_reg7 on page 368
16'hD70	por_cxla_agentid_to_portid_val	RW	5.3.4.36 por_cxla_agentid_to_portid_val on page 369
16'hD78	por_cxla_portfwd_ctl	RW	5.3.4.37 por_cxla_portfwd_ctl on page 370
16'hD80	por_cxla_portfwd_status	RO	5.3.4.38 por_cxla_portfwd_status on page 371
16'h2000	por_cxla_pmu_event_sel	RW	5.3.4.39 por_cxla_pmu_event_sel on page 372
16'h2210	por_cxla_pmu_config	RW	5.3.4.40 por_cxla_pmu_config on page 375
16'h2220	por_cxla_pmevcnt	RW	5.3.4.41 por_cxla_pmevcnt on page 377
16'h2240	por_cxla_pmevcntsr	RW	5.3.4.42 por_cxla_pmevcntsr on page 377

5.2.5 DN register summary

This section lists the DN registers used in CMN-650.

Table 5-6: por_dn_cfg register summary

Offset	Name	Type	Description
16'h0	por_dn_node_info	RO	5.3.5.1 por_dn_node_info on page 378
16'h80	por_dn_child_info	RO	5.3.5.2 por_dn_child_info on page 379
16'h900	por_dn_build_info	RO	5.3.5.3 por_dn_build_info on page 380
16'h980	por_dn_secure_register_groups_override	RW	5.3.5.4 por_dn_secure_register_groups_override on page 381
16'hA00	por_dn_cfg_ctl	RW	5.3.5.5 por_dn_cfg_ctl on page 382
16'hA08	por_dn_aux_ctl	RW	5.3.5.6 por_dn_aux_ctl on page 383
16'hC00 + (56 × #[0, 1, ... 15])	por_dn_vmf0-15_ctrl	RW	5.3.5.7 por_dn_vmf0-15_ctrl on page 384
16'hC00 + (56 × #[0, 1, ... 15] + 8)	por_dn_vmf0-15_rnf0	RW	5.3.5.8 por_dn_vmf0-15_rnf0 on page 385
16'hC00 + (56 × #[0, 1, ... 15] + 16)	por_dn_vmf0-15_rnf1	RW	5.3.5.9 por_dn_vmf0-15_rnf1 on page 386
16'hC00 + (56 × #[0, 1, ... 15] + 24)	por_dn_vmf0-15_rnf2	RW	5.3.5.10 por_dn_vmf0-15_rnf2 on page 387
16'hC00 + (56 × #[0, 1, ... 15] + 32)	por_dn_vmf0-15_rnf3	RW	5.3.5.11 por_dn_vmf0-15_rnf3 on page 388
16'hC00 + (56 × #[0, 1, ... 15] + 40)	por_dn_vmf0-15_rnd	RW	5.3.5.12 por_dn_vmf0-15_rnd on page 389
16'hC00 + (56 × #[0, 1, ... 15] + 48)	por_dn_vmf0-15_cxra	RW	5.3.5.13 por_dn_vmf0-15_cxra on page 390
16'h2000	por_dn_pmu_event_sel	RW	5.3.5.14 por_dn_pmu_event_sel on page 391

5.2.6 CXHA register summary

This section lists the CXHA registers used in CMN-650.

Table 5-7: por_cxg_ha_cfg register summary

Offset	Name	Type	Description
16'h0	por_cxg_ha_node_info	RO	5.3.6.1 por_cxg_ha_node_info on page 392
16'h8	por_cxg_ha_id	RW	5.3.6.2 por_cxg_ha_id on page 393
16'h80	por_cxg_ha_child_info	RO	5.3.6.3 por_cxg_ha_child_info on page 394
16'hA00	por_cxg_ha_cfg_ctl	RW	5.3.6.4 por_cxg_ha_cfg_ctl on page 395
16'hA08	por_cxg_ha_aux_ctl	RW	5.3.6.5 por_cxg_ha_aux_ctl on page 396
16'hA10	por_cxg_ha_mpam_control	RW	5.3.6.6 por_cxg_ha_mpam_control on page 397
16'h980	por_cxg_ha_secure_register_groups_override	RW	5.3.6.7 por_cxg_ha_secure_register_groups_override on page 398
16'h900	por_cxg_ha_unit_info	RO	5.3.6.8 por_cxg_ha_unit_info on page 399
16'h908	por_cxg_ha_unit_info2	RO	5.3.6.9 por_cxg_ha_unit_info2 on page 401

Offset	Name	Type	Description
16'h1F00	por_cxg_ha_agentid_to_linkid_reg0	RW	5.3.6.10 por_cxg_ha_agentid_to_linkid_reg0 on page 401
16'h1F08	por_cxg_ha_agentid_to_linkid_reg1	RW	5.3.6.11 por_cxg_ha_agentid_to_linkid_reg1 on page 403
16'h1F10	por_cxg_ha_agentid_to_linkid_reg2	RW	5.3.6.12 por_cxg_ha_agentid_to_linkid_reg2 on page 404
16'h1F18	por_cxg_ha_agentid_to_linkid_reg3	RW	5.3.6.13 por_cxg_ha_agentid_to_linkid_reg3 on page 406
16'h1F20	por_cxg_ha_agentid_to_linkid_reg4	RW	5.3.6.14 por_cxg_ha_agentid_to_linkid_reg4 on page 407
16'h1F28	por_cxg_ha_agentid_to_linkid_reg5	RW	5.3.6.15 por_cxg_ha_agentid_to_linkid_reg5 on page 408
16'h1F30	por_cxg_ha_agentid_to_linkid_reg6	RW	5.3.6.16 por_cxg_ha_agentid_to_linkid_reg6 on page 410
16'h1F38	por_cxg_ha_agentid_to_linkid_reg7	RW	5.3.6.17 por_cxg_ha_agentid_to_linkid_reg7 on page 411
16'h1FF8	por_cxg_ha_agentid_to_linkid_val	RW	5.3.6.18 por_cxg_ha_agentid_to_linkid_val on page 412
16'hC00 + (8 × #[0, 1, ... 255])	por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255	RW	5.3.6.19 por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255 on page 413
16'h2000	por_cxg_ha_pmu_event_sel	RW	5.3.6.20 por_cxg_ha_pmu_event_sel on page 415
16'h1C00	por_cxg_ha_cxprtcl_link0_ctl	RW	5.3.6.21 por_cxg_ha_cxprtcl_link0_ctl on page 416
16'h1C08	por_cxg_ha_cxprtcl_link0_status	RO	5.3.6.22 por_cxg_ha_cxprtcl_link0_status on page 418
16'h1C10	por_cxg_ha_cxprtcl_link1_ctl	RW	5.3.6.23 por_cxg_ha_cxprtcl_link1_ctl on page 420
16'h1C18	por_cxg_ha_cxprtcl_link1_status	RO	5.3.6.24 por_cxg_ha_cxprtcl_link1_status on page 422
16'h1C20	por_cxg_ha_cxprtcl_link2_ctl	RW	5.3.6.25 por_cxg_ha_cxprtcl_link2_ctl on page 423
16'h1C28	por_cxg_ha_cxprtcl_link2_status	RO	5.3.6.26 por_cxg_ha_cxprtcl_link2_status on page 425
16'h3000	por_cxg_ha_errfr	RO	5.3.6.27 por_cxg_ha_errfr on page 427
16'h3008	por_cxg_ha_errctlr	RW	5.3.6.28 por_cxg_ha_errctlr on page 428
16'h3010	por_cxg_ha_errstatus	W1C	5.3.6.29 por_cxg_ha_errstatus on page 429
16'h3018	por_cxg_ha_erraddr	RW	5.3.6.30 por_cxg_ha_erraddr on page 431
16'h3020	por_cxg_ha_errmisc	RW	5.3.6.31 por_cxg_ha_errmisc on page 432
16'h3100	por_cxg_ha_errfr_NS	RO	5.3.6.32 por_cxg_ha_errfr_NS on page 433
16'h3108	por_cxg_ha_errctlr_NS	RW	5.3.6.33 por_cxg_ha_errctlr_NS on page 434
16'h3110	por_cxg_ha_errstatus_NS	W1C	5.3.6.34 por_cxg_ha_errstatus_NS on page 435
16'h3118	por_cxg_ha_erraddr_NS	RW	5.3.6.35 por_cxg_ha_erraddr_NS on page 436
16'h3120	por_cxg_ha_errmisc_NS	RW	5.3.6.36 por_cxg_ha_errmisc_NS on page 437

5.2.7 RN SAM register summary

This section lists the RN SAM registers used in CMN-650.

Table 5-8: por_rnsam_cfg register summary

Offset	Name	Type	Description
16'h0	por_rnsam_node_info	RO	5.3.7.1 por_rnsam_node_info on page 438
16'h80	por_rnsam_child_info	RO	5.3.7.2 por_rnsam_child_info on page 439
16'h980	por_rnsam_secure_register_groups_override	RW	5.3.7.3 por_rnsam_secure_register_groups_override on page 440
16'h900	por_rnsam_unit_info	RO	5.3.7.4 por_rnsam_unit_info on page 441
16'hC00 + (8 × #[0, 1, ... 19])	non_hash_mem_region_reg0-19	RW	5.3.7.5 non_hash_mem_region_reg0-19 on page 442
16'hD80 + (8 × #[0, 1, ... 4])	non_hash_tgt_nodeid0-4	RW	5.3.7.6 non_hash_tgt_nodeid0-4 on page 444
16'hE00 + (8 × #[0, 1, ... 3])	sys_cache_grp_region0-3	RW	5.3.7.7 sys_cache_grp_region0-3 on page 445
16'hE40 + (8 × #[0, 1, ... 3])	sys_cache_grp_secondary_reg0-3	RW	5.3.7.8 sys_cache_grp_secondary_reg0-3 on page 446
16'hE80	rnsam_hash_addr_mask_reg	RW	5.3.7.9 rnsam_hash_addr_mask_reg on page 448
16'hE90	rnsam_region_cmp_addr_mask_reg	RW	5.3.7.10 rnsam_region_cmp_addr_mask_reg on page 449
16'hEA0	sys_cache_group_hn_count	RW	5.3.7.11 sys_cache_group_hn_count on page 450
16'hEB0	sys_cache_grp_sn_attr	RW	5.3.7.12 sys_cache_grp_sn_attr on page 451
16'hEC0	sys_cache_grp_nonhash_nodeid	RW	5.3.7.13 sys_cache_grp_nonhash_nodeid on page 453
16'hF00 + (8 × #[0, 1, ... 15])	sys_cache_grp_hn_nodeid_reg0-15	RW	5.3.7.14 sys_cache_grp_hn_nodeid_reg0-15 on page 454
16'h1000 + (8 × #[0, 1, ... 15])	sys_cache_grp_sn_nodeid_reg0-15	RW	5.3.7.15 sys_cache_grp_sn_nodeid_reg0-15 on page 455
16'h1100	rnsam_status	RW	5.3.7.16 rnsam_status on page 456
16'h1108	gic_mem_region_reg	RW	5.3.7.17 gic_mem_region_reg on page 457
16'h1120	sys_cache_grp_cal_mode_reg	RW	5.3.7.18 sys_cache_grp_cal_mode_reg on page 459
16'h1140 + (8 × #[0, 1])	sys_cache_grp_sn_sam_cfg0-1	RW	5.3.7.19 sys_cache_grp_sn_sam_cfg0-1 on page 460
16'h1180	sys_cache_grp_hn_cpa_en_reg	RW	5.3.7.20 sys_cache_grp_hn_cpa_en_reg on page 461
16'h1190	sys_cache_grp_hn_cpa_grp_reg	RW	5.3.7.21 sys_cache_grp_hn_cpa_grp_reg on page 462
16'h11A0	cml_port_aggr_mode_ctrl_reg	RW	5.3.7.22 cml_port_aggr_mode_ctrl_reg on page 464
16'h11A8	cml_port_aggr_mode_ctrl_reg1	RW	5.3.7.23 cml_port_aggr_mode_ctrl_reg1 on page 468
16'h11C0 + (8 × #[0, 1, ... 4])	cml_port_aggr_grp0-4_add_mask	RW	5.3.7.24 cml_port_aggr_grp0-4_add_mask on page 472
16'h11F0 + (8 × #[0, 1])	cml_port_aggr_grp_reg0-1	RW	5.3.7.25 cml_port_aggr_grp_reg0-1 on page 472
16'h1208	cml_port_aggr_ctrl_reg	RW	5.3.7.26 cml_port_aggr_ctrl_reg on page 474
16'h1280 + (8 × #[0, 1, ... 7])	sam_qos_mem_region_reg0-7	RW	5.3.7.27 sam_qos_mem_region_reg0-7 on page 476
16'h1400 + (8 × #[0, 1, ... 15])	sys_cache_grp_region0_sn_nodeid_reg0-15	RW	5.3.7.28 sys_cache_grp_region0_sn_nodeid_reg0-15 on page 477

Offset	Name	Type	Description
16'h1500 + (8 × #[0, 1, ... 15])	sys_cache_grp_region1_sn_nodeid_reg0-15	RW	5.3.7.29 sys_cache_grp_region1_sn_nodeid_reg0-15 on page 479

5.2.8 HN-F MPAM_S register summary

This section lists the HN-F MPAM_S registers used in CMN-650.

Table 5-9: por_hnf_mpam_s_cfg register summary

Offset	Name	Type	Description
16'h0	por_hnf_mpam_s_node_info	RO	5.3.8.1 por_hnf_mpam_s_node_info on page 480
16'h80	por_hnf_mpam_s_child_info	RO	5.3.8.2 por_hnf_mpam_s_child_info on page 481
16'h980	por_hnf_mpam_s_secure_register_groups_override	RW	5.3.8.3 por_hnf_mpam_s_secure_register_groups_override on page 481
16'h1008	por_hnf_mpam_sidr	RO	5.3.8.4 por_hnf_mpam_sidr on page 482
16'h10F0	por_hnf_s_mpam_ecr	RW	5.3.8.5 por_hnf_s_mpam_ecr on page 483
16'h10F8	por_hnf_s_mpam_esr	RW	5.3.8.6 por_hnf_s_mpam_esr on page 484
16'h1100	por_hnf_s_mpamcfg_part_sel	RW	5.3.8.7 por_hnf_s_mpamcfg_part_sel on page 485
16'h1108	por_hnf_s_mpamcfg_cmax	RW	5.3.8.8 por_hnf_s_mpamcfg_cmax on page 486
16'h1200	por_hnf_s_mpamcfg_mbw_min	RW	5.3.8.9 por_hnf_s_mpamcfg_mbw_min on page 487
16'h1208	por_hnf_s_mpamcfg_mbw_max	RW	5.3.8.10 por_hnf_s_mpamcfg_mbw_max on page 488
16'h1220	por_hnf_s_mpamcfg_mbw_winwd	RW	5.3.8.11 por_hnf_s_mpamcfg_mbw_winwd on page 490
16'h1400	por_hnf_s_mpamcfg_pri	RW	5.3.8.12 por_hnf_s_mpamcfg_pri on page 491
16'h1500	por_hnf_s_mpamcfg_mbw_prop	RW	5.3.8.13 por_hnf_s_mpamcfg_mbw_prop on page 492
16'h1600	por_hnf_s_mpamcfg_intpartid	RW	5.3.8.14 por_hnf_s_mpamcfg_intpartid on page 493
16'h1800	por_hnf_s_msmon_cfg_mon_sel	RW	5.3.8.15 por_hnf_s_msmon_cfg_mon_sel on page 494
16'h1808	por_hnf_s_msmon_capt_evnt	RW	5.3.8.16 por_hnf_s_msmon_capt_evnt on page 495
16'h1810	por_hnf_s_msmon_cfg_csu_flt	RW	5.3.8.17 por_hnf_s_msmon_cfg_csu_flt on page 496
16'h1818	por_hnf_s_msmon_cfg_csu_ctl	RW	5.3.8.18 por_hnf_s_msmon_cfg_csu_ctl on page 497
16'h1820	por_hnf_s_msmon_cfg_mbwu_flt	RW	5.3.8.19 por_hnf_s_msmon_cfg_mbwu_flt on page 499
16'h1828	por_hnf_s_msmon_cfg_mbwu_ctl	RW	5.3.8.20 por_hnf_s_msmon_cfg_mbwu_ctl on page 500
16'h1840	por_hnf_s_msmon_csu	RW	5.3.8.21 por_hnf_s_msmon_csu on page 502
16'h1848	por_hnf_s_msmon_csu_capture	RW	5.3.8.22 por_hnf_s_msmon_csu_capture on page 503
16'h1860	por_hnf_s_msmon_mbwu	RW	5.3.8.23 por_hnf_s_msmon_mbwu on page 504
16'h1868	por_hnf_s_msmon_mbwu_capture	RW	5.3.8.24 por_hnf_s_msmon_mbwu_capture on page 505
16'h2000	por_hnf_s_mpamcfg_cpbm	RW	5.3.8.25 por_hnf_s_mpamcfg_cpbm on page 506

5.2.9 Configuration master register summary

This section lists the configuration master registers used in CMN-650.

Table 5-10: por_cfgm_cfg register summary

Offset	Name	Type	Description
16'h0	por_cfgm_node_info	RO	5.3.9.1 por_cfgm_node_info on page 507
16'h8	por_cfgm_periph_id_0_periph_id_1	RO	5.3.9.2 por_cfgm_periph_id_0_periph_id_1 on page 508
16'h10	por_cfgm_periph_id_2_periph_id_3	RO	5.3.9.3 por_cfgm_periph_id_2_periph_id_3 on page 509
16'h18	por_cfgm_periph_id_4_periph_id_5	RO	5.3.9.4 por_cfgm_periph_id_4_periph_id_5 on page 510
16'h20	por_cfgm_periph_id_6_periph_id_7	RO	5.3.9.5 por_cfgm_periph_id_6_periph_id_7 on page 511
16'h28	por_cfgm_component_id_0_component_id_1	RO	5.3.9.6 por_cfgm_component_id_0_component_id_1 on page 512
16'h30	por_cfgm_component_id_2_component_id_3	RO	5.3.9.7 por_cfgm_component_id_2_component_id_3 on page 513
16'h80	por_cfgm_child_info	RO	5.3.9.8 por_cfgm_child_info on page 514
16'h980	por_cfgm_secure_access	RW	5.3.9.9 por_cfgm_secure_access on page 515
16'h3000 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_mxp_0-7	RO	5.3.9.10 por_cfgm_errgsr_mxp_0-7 on page 516
16'h3040 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_mxp_0-7_NS	RO	5.3.9.11 por_cfgm_errgsr_mxp_0-7_NS on page 517
16'h3080 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_hni_0-7	RO	5.3.9.12 por_cfgm_errgsr_hni_0-7 on page 518
16'h30C0 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_hni_0-7_NS	RO	5.3.9.13 por_cfgm_errgsr_hni_0-7_NS on page 518
16'h3100 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_hnf_0-7	RO	5.3.9.14 por_cfgm_errgsr_hnf_0-7 on page 519
16'h3140 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_hnf_0-7_NS	RO	5.3.9.15 por_cfgm_errgsr_hnf_0-7_NS on page 520
16'h3180 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_sbsx_0-7	RO	5.3.9.16 por_cfgm_errgsr_sbsx_0-7 on page 521
16'h31C0 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_sbsx_0-7_NS	RO	5.3.9.17 por_cfgm_errgsr_sbsx_0-7_NS on page 522
16'h3200 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_cxg_0-7	RO	5.3.9.18 por_cfgm_errgsr_cxg_0-7 on page 523
16'h3240 + (8 × #[0, 1, ... 7])	por_cfgm_errgsr_cxg_0-7_NS	RO	5.3.9.19 por_cfgm_errgsr_cxg_0-7_NS on page 523
16'h3FA8	por_cfgm_errdevaff	RO	5.3.9.20 por_cfgm_errdevaff on page 524
16'h3FB8	por_cfgm_errdevarch	RO	5.3.9.21 por_cfgm_errdevarch on page 525
16'h3FC8	por_cfgm_erridr	RO	5.3.9.22 por_cfgm_erridr on page 526
16'h3FD0	por_cfgm_errpidr45	RO	5.3.9.23 por_cfgm_errpidr45 on page 527
16'h3FD8	por_cfgm_errpidr67	RO	5.3.9.24 por_cfgm_errpidr67 on page 528
16'h3FE0	por_cfgm_errpidr01	RO	5.3.9.25 por_cfgm_errpidr01 on page 529
16'h3FE8	por_cfgm_errpidr23	RO	5.3.9.26 por_cfgm_errpidr23 on page 529
16'h3FF0	por_cfgm_errcidr01	RO	5.3.9.27 por_cfgm_errcidr01 on page 530

Offset	Name	Type	Description
16'h3FF8	por_cfgm_errcidr23	RO	5.3.9.28 por_cfgm_errcidr23 on page 531
16'h900	por_info_global	RO	5.3.9.29 por_info_global on page 532
16'h1C00	por_ppu_int_enable	RW	5.3.9.30 por_ppu_int_enable on page 534
16'h1C08	por_ppu_int_status	W1C	5.3.9.31 por_ppu_int_status on page 534
16'h1C10	por_ppu_qactive_hyst	RW	5.3.9.32 por_ppu_qactive_hyst on page 535
16'h1C18	por_mpam_s_err_int_status	W1C	5.3.9.33 por_mpam_s_err_int_status on page 536
16'h1C20	por_mpam_ns_err_int_status	W1C	5.3.9.34 por_mpam_ns_err_int_status on page 537
16'h100 + (8 × #[0, 1, ... 255])	por_cfgm_child_pointer_0-255	RO	5.3.9.35 por_cfgm_child_pointer_0-255 on page 538

5.2.10 Debug and trace register summary

This section lists the debug and trace registers used in CMN-650.

Table 5-11: por_dt_cfg register summary

Offset	Name	Type	Description
16'h0	por_dt_node_info	RO	5.3.10.1 por_dt_node_info on page 539
16'h80	por_dt_child_info	RO	5.3.10.2 por_dt_child_info on page 540
16'h980	por_dt_secure_access	RW	5.3.10.3 por_dt_secure_access on page 540
16'hA00	por_dt_dtc_ctl	RW	5.3.10.4 por_dt_dtc_ctl on page 542
16'hA10	por_dt_trigger_status	RO	5.3.10.5 por_dt_trigger_status on page 543
16'hA20	por_dt_trigger_status_clr	WO	5.3.10.6 por_dt_trigger_status_clr on page 544
16'hA30	por_dt_trace_control	RW	5.3.10.7 por_dt_trace_control on page 545
16'hA48	por_dt_traceid	RW	5.3.10.8 por_dt_traceid on page 546
16'h2000	por_dt_pmevcntAB	RW	5.3.10.9 por_dt_pmevcntAB on page 547
16'h2010	por_dt_pmevcntCD	RW	5.3.10.10 por_dt_pmevcntCD on page 547
16'h2020	por_dt_pmevcntEF	RW	5.3.10.11 por_dt_pmevcntEF on page 548
16'h2030	por_dt_pmevcntGH	RW	5.3.10.12 por_dt_pmevcntGH on page 549
16'h2040	por_dt_pmccntr	RW	5.3.10.13 por_dt_pmccntr on page 550
16'h2050	por_dt_pmevcntsrAB	RW	5.3.10.14 por_dt_pmevcntsrAB on page 551
16'h2060	por_dt_pmevcntsrCD	RW	5.3.10.15 por_dt_pmevcntsrCD on page 552
16'h2070	por_dt_pmevcntsrEF	RW	5.3.10.16 por_dt_pmevcntsrEF on page 552
16'h2080	por_dt_pmevcntsrGH	RW	5.3.10.17 por_dt_pmevcntsrGH on page 553
16'h2090	por_dt_pmccntrsr	RW	5.3.10.18 por_dt_pmccntrsr on page 554
16'h2100	por_dt_pmcr	RW	5.3.10.19 por_dt_pmcr on page 555
16'h2118	por_dt_pmovsr	RO	5.3.10.20 por_dt_pmovsr on page 556
16'h2120	por_dt_pmovsr_clr	WO	5.3.10.21 por_dt_pmovsr_clr on page 557
16'h2128	por_dt_pmssr	RO	5.3.10.22 por_dt_pmssr on page 558
16'h2130	por_dt_pmsrr	WO	5.3.10.23 por_dt_pmsrr on page 559
16'hFA0	por_dt_claim	RW	5.3.10.24 por_dt_claim on page 560

Offset	Name	Type	Description
16'hFA8	por_dt_devaff	RO	5.3.10.25 por_dt_devaff on page 560
16'hFB0	por_dt_lsr	RO	5.3.10.26 por_dt_lsr on page 561
16'hFB8	por_dt_authstatus_devarch	RO	5.3.10.27 por_dt_authstatus_devarch on page 562
16'hFC0	por_dt_devid	RO	5.3.10.28 por_dt_devid on page 563
16'hFC8	por_dt_devtype	RO	5.3.10.29 por_dt_devtype on page 564
16'hFD0	por_dt_pidr45	RO	5.3.10.30 por_dt_pidr45 on page 565
16'hFD8	por_dt_pidr67	RO	5.3.10.31 por_dt_pidr67 on page 566
16'hFE0	por_dt_pidr01	RO	5.3.10.32 por_dt_pidr01 on page 567
16'hFE8	por_dt_pidr23	RO	5.3.10.33 por_dt_pidr23 on page 568
16'hFF0	por_dt_cidr01	RO	5.3.10.34 por_dt_cidr01 on page 569
16'hFF8	por_dt_cidr23	RO	5.3.10.35 por_dt_cidr23 on page 570

5.2.11 SBSX register summary

This section lists the SBSX registers used in CMN-650.

Table 5-12: por_sbsx_cfg register summary

Offset	Name	Type	Description
16'h0	por_sbsx_node_info	RO	5.3.11.1 por_sbsx_node_info on page 571
16'h80	por_sbsx_child_info	RO	5.3.11.2 por_sbsx_child_info on page 572
16'h980	por_sbsx_secure_register_groups_override	RW	5.3.11.3 por_sbsx_secure_register_groups_override on page 573
16'h900	por_sbsx_unit_info	RO	5.3.11.4 por_sbsx_unit_info on page 574
16'hA00	por_sbsx_cfg_ctl	RW	5.3.11.5 por_sbsx_cfg_ctl on page 576
16'hA08	por_sbsx_aux_ctl	RW	5.3.11.6 por_sbsx_aux_ctl on page 577
16'hA18	por_sbsx_cbusy_limit_ctl	RW	5.3.11.7 por_sbsx_cbusy_limit_ctl on page 577
16'h3000	por_sbsx_errfr	RO	5.3.11.8 por_sbsx_errfr on page 578
16'h3008	por_sbsx_errctlr	RW	5.3.11.9 por_sbsx_errctlr on page 579
16'h3010	por_sbsx_errstatus	W1C	5.3.11.10 por_sbsx_errstatus on page 580
16'h3018	por_sbsx_erraddr	RW	5.3.11.11 por_sbsx_erraddr on page 582
16'h3020	por_sbsx_errmisc	RW	5.3.11.12 por_sbsx_errmisc on page 583
16'h3100	por_sbsx_errfr_NS	RO	5.3.11.13 por_sbsx_errfr_NS on page 584
16'h3108	por_sbsx_errctlr_NS	RW	5.3.11.14 por_sbsx_errctlr_NS on page 585
16'h3110	por_sbsx_errstatus_NS	W1C	5.3.11.15 por_sbsx_errstatus_NS on page 586
16'h3118	por_sbsx_erraddr_NS	RW	5.3.11.16 por_sbsx_erraddr_NS on page 588
16'h3120	por_sbsx_errmisc_NS	RW	5.3.11.17 por_sbsx_errmisc_NS on page 589
16'h2000	por_sbsx_pmu_event_sel	RW	5.3.11.18 por_sbsx_pmu_event_sel on page 590

5.2.12 HN-F MPAM_NS register summary

This section lists the HN-F MPAM_NS registers used in CMN-650.

Table 5-13: por_hnf_mpam_ns_cfg register summary

Offset	Name	Type	Description
16'h0	por_hnf_mpam_ns_node_info	RO	5.3.12.1 por_hnf_mpam_ns_node_info on page 592
16'h80	por_hnf_mpam_ns_child_info	RO	5.3.12.2 por_hnf_mpam_ns_child_info on page 593
16'h1000	por_hnf_mpam_idr	RO	5.3.12.3 por_hnf_mpam_idr on page 594
16'h1018	por_hnf_mpam_iidr	RO	5.3.12.4 por_hnf_mpam_iidr on page 596
16'h1020	por_hnf_mpam_aidr	RO	5.3.12.5 por_hnf_mpam_aidr on page 596
16'h1028	por_hnf_mpam_impl_idr	RO	5.3.12.6 por_hnf_mpam_impl_idr on page 597
16'h1030	por_hnf_mpam_cpor_idr	RO	5.3.12.7 por_hnf_mpam_cpor_idr on page 598
16'h1038	por_hnf_mpam_ccap_idr	RO	5.3.12.8 por_hnf_mpam_ccap_idr on page 599
16'h1040	por_hnf_mpam_mbw_idr	RO	5.3.12.9 por_hnf_mpam_mbw_idr on page 600
16'h1048	por_hnf_mpam_pri_idr	RO	5.3.12.10 por_hnf_mpam_pri_idr on page 601
16'h1050	por_hnf_mpam_partid_nrw_idr	RO	5.3.12.11 por_hnf_mpam_partid_nrw_idr on page 603
16'h1080	por_hnf_mpam_msmon_idr	RO	5.3.12.12 por_hnf_mpam_msmon_idr on page 604
16'h1088	por_hnf_mpam_csumon_idr	RO	5.3.12.13 por_hnf_mpam_csumon_idr on page 605
16'h1090	por_hnf_mpam_mbwumon_idr	RO	5.3.12.14 por_hnf_mpam_mbwumon_idr on page 606
16'h10F0	por_hnf_ns_mpam_ecr	RW	5.3.12.15 por_hnf_ns_mpam_ecr on page 607
16'h10F8	por_hnf_ns_mpam_esr	RW	5.3.12.16 por_hnf_ns_mpam_esr on page 608
16'h1100	por_hnf_ns_mpamcfg_part_sel	RW	5.3.12.17 por_hnf_ns_mpamcfg_part_sel on page 609
16'h1108	por_hnf_ns_mpamcfg_cmax	RW	5.3.12.18 por_hnf_ns_mpamcfg_cmax on page 610
16'h1200	por_hnf_ns_mpamcfg_mbw_min	RW	5.3.12.19 por_hnf_ns_mpamcfg_mbw_min on page 611
16'h1208	por_hnf_ns_mpamcfg_mbw_max	RW	5.3.12.20 por_hnf_ns_mpamcfg_mbw_max on page 612
16'h1220	por_hnf_ns_mpamcfg_mbw_winwd	RW	5.3.12.21 por_hnf_ns_mpamcfg_mbw_winwd on page 613
16'h1400	por_hnf_ns_mpamcfg_pri	RW	5.3.12.22 por_hnf_ns_mpamcfg_pri on page 614
16'h1500	por_hnf_ns_mpamcfg_mbw_prop	RW	5.3.12.23 por_hnf_ns_mpamcfg_mbw_prop on page 615
16'h1600	por_hnf_ns_mpamcfg_intpartid	RW	5.3.12.24 por_hnf_ns_mpamcfg_intpartid on page 616
16'h1800	por_hnf_ns_msmon_cfg_mon_sel	RW	5.3.12.25 por_hnf_ns_msmon_cfg_mon_sel on page 617
16'h1808	por_hnf_ns_msmon_capt_evnt	RW	5.3.12.26 por_hnf_ns_msmon_capt_evnt on page 618
16'h1810	por_hnf_ns_msmon_cfg_csu_flt	RW	5.3.12.27 por_hnf_ns_msmon_cfg_csu_flt on page 619
16'h1818	por_hnf_ns_msmon_cfg_csu_ctl	RW	5.3.12.28 por_hnf_ns_msmon_cfg_csu_ctl on page 620
16'h1820	por_hnf_ns_msmon_cfg_mbwu_flt	RW	5.3.12.29 por_hnf_ns_msmon_cfg_mbwu_flt on page 622
16'h1828	por_hnf_ns_msmon_cfg_mbwu_ctl	RW	5.3.12.30 por_hnf_ns_msmon_cfg_mbwu_ctl on page 623
16'h1840	por_hnf_ns_msmon_csu	RW	5.3.12.31 por_hnf_ns_msmon_csu on page 625
16'h1848	por_hnf_ns_msmon_csu_capture	RW	5.3.12.32 por_hnf_ns_msmon_csu_capture on page 626
16'h1860	por_hnf_ns_msmon_mbwu	RW	5.3.12.33 por_hnf_ns_msmon_mbwu on page 627
16'h1868	por_hnf_ns_msmon_mbwu_capture	RW	5.3.12.34 por_hnf_ns_msmon_mbwu_capture on page 628
16'h2000	por_hnf_ns_mpamcfg_cpbm	RW	5.3.12.35 por_hnf_ns_mpamcfg_cpbm on page 629

5.2.13 RN-I register summary

This section lists the RN-I registers used in CMN-650.

Table 5-14: por_rni_cfg register summary

Offset	Name	Type	Description
16'h0	por_rni_node_info	RO	5.3.13.1 por_rni_node_info on page 630
16'h80	por_rni_child_info	RO	5.3.13.2 por_rni_child_info on page 631
16'h980	por_rni_secure_register_groups_override	RW	5.3.13.3 por_rni_secure_register_groups_override on page 632
16'h900	por_rni_unit_info	RO	5.3.13.4 por_rni_unit_info on page 633
16'h908	por_rni_unit_info2	RO	5.3.13.5 por_rni_unit_info2 on page 634
16'hA00	por_rni_cfg_ctl	RW	5.3.13.6 por_rni_cfg_ctl on page 635
16'hA08	por_rni_aux_ctl	RW	5.3.13.7 por_rni_aux_ctl on page 637
16'hA10 + (8 × #[0, 1, 2])	por_rni_s0-2_port_control	RW	5.3.13.8 por_rni_s0-2_port_control on page 639
16'hA28 + (8 × #[0, 1, 2])	por_rni_s0-2_mpam_control	RW	5.3.13.9 por_rni_s0-2_mpam_control on page 640
16'hA80 + (32 × #[0, 1, 2])	por_rni_s0-2_qos_control	RW	5.3.13.10 por_rni_s0-2_qos_control on page 641
16'hA88 + (32 × #[0, 1, 2])	por_rni_s0-2_qos_lat_tgt	RW	5.3.13.11 por_rni_s0-2_qos_lat_tgt on page 643
16'hA90 + (32 × #[0, 1, 2])	por_rni_s0-2_qos_lat_scale	RW	5.3.13.12 por_rni_s0-2_qos_lat_scale on page 644
16'hA98 + (32 × #[0, 1, 2])	por_rni_s0-2_qos_lat_range	RW	5.3.13.13 por_rni_s0-2_qos_lat_range on page 646
16'h2000	por_rni_pmu_event_sel	RW	5.3.13.14 por_rni_pmu_event_sel on page 647

5.2.14 XP register summary

This section lists the XP registers used in CMN-650.

Table 5-15: por_mxp_cfg register summary

Offset	Name	Type	Description
16'h0	por_mxp_node_info	RO	5.3.14.1 por_mxp_node_info on page 650
16'h8 + (8 × #[0, 1])	por_mxp_device_port_connect_info_p0-1	RO	5.3.14.2 por_mxp_device_port_connect_info_p0-1 on page 650
16'h18	por_mxp_mesh_port_connect_info_east	RO	5.3.14.3 por_mxp_mesh_port_connect_info_east on page 654
16'h20	por_mxp_mesh_port_connect_info_north	RO	5.3.14.4 por_mxp_mesh_port_connect_info_north on page 654
16'h80	por_mxp_child_info	RO	5.3.14.5 por_mxp_child_info on page 655
16'h100 + (8 × #[0, 1, ... 31])	por_mxp_child_pointer_0-31	RO	5.3.14.6 por_mxp_child_pointer_0-31 on page 656
16'h900 + (8 × #[0, 1])	por_mxp_p0-1_info	RO	5.3.14.7 por_mxp_p0-1_info on page 657

Offset	Name	Type	Description
16'h910	por_dtm_unit_info	RO	5.3.14.8 por_dtm_unit_info on page 658
16'h980	por_mxp_secure_register_groups_override	RW	5.3.14.9 por_mxp_secure_register_groups_override on page 659
16'hA00	por_mxp_aux_ctl	RW	5.3.14.10 por_mxp_aux_ctl on page 660
16'hA08 + (8 × #[0, 1])	por_mxp_p0-1_mpam_override	RW	5.3.14.11 por_mxp_p0-1_mpam_override on page 661
16'hA38 + (8 × #[0, 1])	por_mxp_p0-1_ldid_override	RW	5.3.14.12 por_mxp_p0-1_ldid_override on page 662
16'hA80 + (32 × #[0, 1])	por_mxp_p0-1_qos_control	RW	5.3.14.13 por_mxp_p0-1_qos_control on page 664
16'hA88 + (32 × #[0, 1])	por_mxp_p0-1_qos_lat_tgt	RW	5.3.14.14 por_mxp_p0-1_qos_lat_tgt on page 665
16'hA90 + (32 × #[0, 1])	por_mxp_p0-1_qos_lat_scale	RW	5.3.14.15 por_mxp_p0-1_qos_lat_scale on page 666
16'hA98 + (32 × #[0, 1])	por_mxp_p0-1_qos_lat_range	RW	5.3.14.16 por_mxp_p0-1_qos_lat_range on page 667
16'h2000	por_mxp_pmu_event_sel	RW	5.3.14.17 por_mxp_pmu_event_sel on page 668
16'h3000	por_mxp_errfr	RO	5.3.14.18 por_mxp_errfr on page 671
16'h3008	por_mxp_errctlr	RW	5.3.14.19 por_mxp_errctlr on page 671
16'h3010	por_mxp_errstatus	W1C	5.3.14.20 por_mxp_errstatus on page 672
16'h3028	por_mxp_errmisc	RW	5.3.14.21 por_mxp_errmisc on page 674
16'h3030 + (8 × #[0, 1])	por_mxp_p0-1_byte_par_err_inj	WO	5.3.14.22 por_mxp_p0-1_byte_par_err_inj on page 675
16'h3100	por_mxp_errfr_NS	RO	5.3.14.23 por_mxp_errfr_NS on page 676
16'h3108	por_mxp_errctlr_NS	RW	5.3.14.24 por_mxp_errctlr_NS on page 677
16'h3110	por_mxp_errstatus_NS	W1C	5.3.14.25 por_mxp_errstatus_NS on page 678
16'h3128	por_mxp_errmisc_NS	RW	5.3.14.26 por_mxp_errmisc_NS on page 680
16'h1C00 + (8 × #[0, 1])	por_mxp_p0-1_syscoreq_ctl	RW	5.3.14.27 por_mxp_p0-1_syscoreq_ctl on page 681
16'h1C10 + (8 × #[0, 1])	por_mxp_p0-1_syscoack_status	RO	5.3.14.28 por_mxp_p0-1_syscoack_status on page 683
16'h2100	por_dtm_control	RW	5.3.14.29 por_dtm_control on page 684
16'h2118	por_dtm_fifo_entry_ready	W1C	5.3.14.30 por_dtm_fifo_entry_ready on page 685
16'h2120 + (24 × #[0, 1, ... 3])	por_dtm_fifo_entry0-3_0	RO	5.3.14.31 por_dtm_fifo_entry0-3_0 on page 686
16'h2128 + (24 × #[0, 1, ... 3])	por_dtm_fifo_entry0-3_1	RO	5.3.14.32 por_dtm_fifo_entry0-3_1 on page 687
16'h2130 + (24 × #[0, 1, ... 3])	por_dtm_fifo_entry0-3_2	RO	5.3.14.33 por_dtm_fifo_entry0-3_2 on page 688
16'h21A0 + (24 × #[0, 1, ... 3])	por_dtm_wp0-3_config	RW	5.3.14.34 por_dtm_wp0-3_config on page 688
16'h21A8 + (24 × #[0, 1, ... 3])	por_dtm_wp0-3_val	RW	5.3.14.35 por_dtm_wp0-3_val on page 691
16'h21B0 + (24 × #[0, 1, ... 3])	por_dtm_wp0-3_mask	RW	5.3.14.36 por_dtm_wp0-3_mask on page 692
16'h2200	por_dtm_pmsicr	RW	5.3.14.37 por_dtm_pmsicr on page 693
16'h2208	por_dtm_pmsirr	RW	5.3.14.38 por_dtm_pmsirr on page 693
16'h2210	por_dtm_pmu_config	RW	5.3.14.39 por_dtm_pmu_config on page 694
16'h2220	por_dtm_pmevcnt	RW	5.3.14.40 por_dtm_pmevcnt on page 700
16'h2240	por_dtm_pmevcntsr	RW	5.3.14.41 por_dtm_pmevcntsr on page 701
16'hC00 + (8 × #[0, 1, ... 15])	por_mxp_multi_dat_rsp_chn_sel_0-15	RW	5.3.14.42 por_mxp_multi_dat_rsp_chn_sel_0-15 on page 702

Offset	Name	Type	Description
16'hC80	por_mxp_multi_dat_rsp_chn_ctrl	RW	5.3.14.43 por_mxp_multi_dat_rsp_chn_ctrl on page 704
16'hC90 + (8 × #[0, 1, ... 7])	por_mxp_xy_override_sel_0-7	RW	5.3.14.44 por_mxp_xy_override_sel_0-7 on page 705

5.2.15 HN-F register summary

This section lists the HN-F registers used in CMN-650.

Table 5-16: por_hnf_cfg register summary

Offset	Name	Type	Description
16'h0	por_hnf_node_info	RO	5.3.15.1 por_hnf_node_info on page 707
16'h80	por_hnf_child_info	RO	5.3.15.2 por_hnf_child_info on page 708
16'h980	por_hnf_secure_register_groups_override	RW	5.3.15.3 por_hnf_secure_register_groups_override on page 709
16'h900	por_hnf_unit_info	RO	5.3.15.4 por_hnf_unit_info on page 710
16'h908	por_hnf_unit_info_1	RO	5.3.15.5 por_hnf_unit_info_1 on page 713
16'hA00	por_hnf_cfg_ctl	RW	5.3.15.6 por_hnf_cfg_ctl on page 714
16'hA08	por_hnf_aux_ctl	RW	5.3.15.7 por_hnf_aux_ctl on page 716
16'hA10	por_hnf_r2_aux_ctl	RW	5.3.15.8 por_hnf_r2_aux_ctl on page 720
16'hA18	por_hnf_cbusy_limit_ctl	RW	5.3.15.9 por_hnf_cbusy_limit_ctl on page 722
16'h1C00	por_hnf_ppu_pwpr	RW	5.3.15.10 por_hnf_ppu_pwpr on page 723
16'h1C08	por_hnf_ppu_pwsr	RO	5.3.15.11 por_hnf_ppu_pwsr on page 724
16'h1C14	por_hnf_ppu_misr	RO	5.3.15.12 por_hnf_ppu_misr on page 725
16'h2BB0	por_hnf_ppu_idr0	RO	5.3.15.13 por_hnf_ppu_idr0 on page 726
16'h2BB4	por_hnf_ppu_idr1	RO	5.3.15.14 por_hnf_ppu_idr1 on page 728
16'h2BC8	por_hnf_ppu_iidr	RO	5.3.15.15 por_hnf_ppu_iidr on page 729
16'h2BCC	por_hnf_ppu_aidr	RO	5.3.15.16 por_hnf_ppu_aidr on page 729
16'h1D00	por_hnf_ppu_dyn_ret_threshold	RW	5.3.15.17 por_hnf_ppu_dyn_ret_threshold on page 730
16'hA80	por_hnf_qos_band	RO	5.3.15.18 por_hnf_qos_band on page 731
16'hA88	por_hnf_qos_reservation	RW	5.3.15.19 por_hnf_qos_reservation on page 732
16'hA90	por_hnf_rn_starvation	RW	5.3.15.20 por_hnf_rn_starvation on page 734
16'h3000	por_hnf_errfr	RO	5.3.15.21 por_hnf_errfr on page 735
16'h3008	por_hnf_errctlr	RW	5.3.15.22 por_hnf_errctlr on page 736
16'h3010	por_hnf_errstatus	W1C	5.3.15.23 por_hnf_errstatus on page 737
16'h3018	por_hnf_erraddr	RW	5.3.15.24 por_hnf_erraddr on page 739
16'h3020	por_hnf_errmisc	RW	5.3.15.25 por_hnf_errmisc on page 740
16'h3030	por_hnf_err_inj	RW	5.3.15.26 por_hnf_err_inj on page 742
16'h3038	por_hnf_byte_par_err_inj	WO	5.3.15.27 por_hnf_byte_par_err_inj on page 743
16'h3100	por_hnf_errfr_NS	RO	5.3.15.28 por_hnf_errfr_NS on page 744
16'h3108	por_hnf_errctlr_NS	RW	5.3.15.29 por_hnf_errctlr_NS on page 745

Offset	Name	Type	Description
16'h3110	por_hnf_errstatus_NS	W1C	5.3.15.30 por_hnf_errstatus_NS on page 746
16'h3118	por_hnf_erraddr_NS	RW	5.3.15.31 por_hnf_erraddr_NS on page 748
16'h3120	por_hnf_errmisc_NS	RW	5.3.15.32 por_hnf_errmisc_NS on page 749
16'hC00	por_hnf_slc_lock_ways	RW	5.3.15.33 por_hnf_slc_lock_ways on page 750
16'hC08	por_hnf_slc_lock_base0	RW	5.3.15.34 por_hnf_slc_lock_base0 on page 751
16'hC10	por_hnf_slc_lock_base1	RW	5.3.15.35 por_hnf_slc_lock_base1 on page 752
16'hC18	por_hnf_slc_lock_base2	RW	5.3.15.36 por_hnf_slc_lock_base2 on page 753
16'hC20	por_hnf_slc_lock_base3	RW	5.3.15.37 por_hnf_slc_lock_base3 on page 754
16'hC28	por_hnf_rni_region_vec	RW	5.3.15.38 por_hnf_rni_region_vec on page 755
16'hC30	por_hnf_rnd_region_vec	RW	5.3.15.39 por_hnf_rnd_region_vec on page 756
16'hC38	por_hnf_rnf_region_vec	RW	5.3.15.40 por_hnf_rnf_region_vec on page 757
16'hC40	por_hnf_rnf_region_vec1	RW	5.3.15.41 por_hnf_rnf_region_vec1 on page 758
16'hC48	por_hnf_slcway_partition0_rnf_vec	RW	5.3.15.42 por_hnf_slcway_partition0_rnf_vec on page 759
16'hC50	por_hnf_slcway_partition1_rnf_vec	RW	5.3.15.43 por_hnf_slcway_partition1_rnf_vec on page 760
16'hC58	por_hnf_slcway_partition2_rnf_vec	RW	5.3.15.44 por_hnf_slcway_partition2_rnf_vec on page 761
16'hC60	por_hnf_slcway_partition3_rnf_vec	RW	5.3.15.45 por_hnf_slcway_partition3_rnf_vec on page 762
16'hCB0	por_hnf_slcway_partition0_rnf_vec1	RW	5.3.15.46 por_hnf_slcway_partition0_rnf_vec1 on page 763
16'hCB8	por_hnf_slcway_partition1_rnf_vec1	RW	5.3.15.47 por_hnf_slcway_partition1_rnf_vec1 on page 764
16'hCC0	por_hnf_slcway_partition2_rnf_vec1	RW	5.3.15.48 por_hnf_slcway_partition2_rnf_vec1 on page 765
16'hCC8	por_hnf_slcway_partition3_rnf_vec1	RW	5.3.15.49 por_hnf_slcway_partition3_rnf_vec1 on page 766
16'hC68	por_hnf_slcway_partition0_rni_vec	RW	5.3.15.50 por_hnf_slcway_partition0_rni_vec on page 767
16'hC70	por_hnf_slcway_partition1_rni_vec	RW	5.3.15.51 por_hnf_slcway_partition1_rni_vec on page 768
16'hC78	por_hnf_slcway_partition2_rni_vec	RW	5.3.15.52 por_hnf_slcway_partition2_rni_vec on page 769
16'hC80	por_hnf_slcway_partition3_rni_vec	RW	5.3.15.53 por_hnf_slcway_partition3_rni_vec on page 770
16'hC88	por_hnf_slcway_partition0_rnd_vec	RW	5.3.15.54 por_hnf_slcway_partition0_rnd_vec on page 771
16'hC90	por_hnf_slcway_partition1_rnd_vec	RW	5.3.15.55 por_hnf_slcway_partition1_rnd_vec on page 772
16'hC98	por_hnf_slcway_partition2_rnd_vec	RW	5.3.15.56 por_hnf_slcway_partition2_rnd_vec on page 773
16'hCA0	por_hnf_slcway_partition3_rnd_vec	RW	5.3.15.57 por_hnf_slcway_partition3_rnd_vec on page 774
16'hCA8	por_hnf_rn_region_lock	RW	5.3.15.58 por_hnf_rn_region_lock on page 775

Offset	Name	Type	Description
16'hCD0	por_hnf_sf_cxg_blocked_ways	RW	5.3.15.59 por_hnf_sf_cxg_blocked_ways on page 776
16'hCE0	por_hnf_cxg_ha_metadata_exclusion_list	RW	5.3.15.60 por_hnf_cxg_ha_metadata_exclusion_list on page 777
16'hCF0	hn_sam_hash_addr_mask_reg	RW	5.3.15.61 hn_sam_hash_addr_mask_reg on page 778
16'hCF8	hn_sam_region_cmp_addr_mask_reg	RW	5.3.15.62 hn_sam_region_cmp_addr_mask_reg on page 779
16'hD00	por_hnf_sam_control	RW	5.3.15.63 por_hnf_sam_control on page 780
16'hD08	por_hnf_sam_memregion0	RW	5.3.15.64 por_hnf_sam_memregion0 on page 781
16'hD10	por_hnf_sam_memregion1	RW	5.3.15.65 por_hnf_sam_memregion1 on page 783
16'hD18	por_hnf_sam_sn_properties	RW	5.3.15.66 por_hnf_sam_sn_properties on page 784
16'hD20	por_hnf_sam_6sn_nodeid	RW	5.3.15.67 por_hnf_sam_6sn_nodeid on page 787
16'hCE8	por_hnf_sam_sn_properties1	RW	5.3.15.68 por_hnf_sam_sn_properties1 on page 788
16'hF80	por_hnf_cml_port_aggr_grp0_add_mask	RW	5.3.15.69 por_hnf_cml_port_aggr_grp0_add_mask on page 791
16'hF88	por_hnf_cml_port_aggr_grp1_add_mask	RW	5.3.15.70 por_hnf_cml_port_aggr_grp1_add_mask on page 792
16'hF90	por_hnf_cml_port_aggr_grp2_add_mask	RW	5.3.15.71 por_hnf_cml_port_aggr_grp2_add_mask on page 793
16'hF98	por_hnf_cml_port_aggr_grp3_add_mask	RW	5.3.15.72 por_hnf_cml_port_aggr_grp3_add_mask on page 794
16'hFA0	por_hnf_cml_port_aggr_grp4_add_mask	RW	5.3.15.73 por_hnf_cml_port_aggr_grp4_add_mask on page 795
16'hFB0	por_hnf_cml_port_aggr_grp_reg0	RW	5.3.15.74 por_hnf_cml_port_aggr_grp_reg0 on page 796
16'hFB8	por_hnf_cml_port_aggr_grp_reg1	RW	5.3.15.75 por_hnf_cml_port_aggr_grp_reg1 on page 797
16'hFD0	por_hnf_cml_port_aggr_ctrl_reg	RW	5.3.15.76 por_hnf_cml_port_aggr_ctrl_reg on page 798
16'hF50	por_hnf_abf_lo_addr	RW	5.3.15.77 por_hnf_abf_lo_addr on page 800
16'hF58	por_hnf_abf_hi_addr	RW	5.3.15.78 por_hnf_abf_hi_addr on page 801
16'hF60	por_hnf_abf_pr	RW	5.3.15.79 por_hnf_abf_pr on page 802
16'hF68	por_hnf_abf_sr	RO	5.3.15.80 por_hnf_abf_sr on page 803
16'h1000	por_hnf_cbusy_write_limit_ctl	RW	5.3.15.81 por_hnf_cbusy_write_limit_ctl on page 804
16'h1008	por_hnf_cbusy_resp_ctl	RW	5.3.15.82 por_hnf_cbusy_resp_ctl on page 805
16'h1010	por_hnf_cbusy_sn_ctl	RW	5.3.15.83 por_hnf_cbusy_sn_ctl on page 806
16'hFE0	por_hnf_partner_scratch_reg0	RW	5.3.15.84 por_hnf_partner_scratch_reg0 on page 808
16'hFE8	por_hnf_partner_scratch_reg1	RW	5.3.15.85 por_hnf_partner_scratch_reg1 on page 808
16'hB80	por_hnf_cfg_slcsf_dbgrd	WO	5.3.15.86 por_hnf_cfg_slcsf_dbgrd on page 809
16'hB88	por_hnf_slc_cache_access_slc_tag	RO	5.3.15.87 por_hnf_slc_cache_access_slc_tag on page 810
16'hB90	por_hnf_slc_cache_access_slc_tag1	RO	5.3.15.88 por_hnf_slc_cache_access_slc_tag1 on page 811
16'hB98	por_hnf_slc_cache_access_slc_data	RO	5.3.15.89 por_hnf_slc_cache_access_slc_data on page 812

Offset	Name	Type	Description
16'hBC0	por_hnf_slc_cache_access_slc_mte_tag	RO	5.3.15.90 por_hnf_slc_cache_access_slc_mte_tag on page 813
16'hBA0	por_hnf_slc_cache_access_sf_tag	RO	5.3.15.91 por_hnf_slc_cache_access_sf_tag on page 814
16'hBA8	por_hnf_slc_cache_access_sf_tag1	RO	5.3.15.92 por_hnf_slc_cache_access_sf_tag1 on page 815
16'hBB0	por_hnf_slc_cache_access_sf_tag2	RO	5.3.15.93 por_hnf_slc_cache_access_sf_tag2 on page 816
16'h2000	por_hnf_pmu_event_sel	RW	5.3.15.94 por_hnf_pmu_event_sel on page 816
16'h2008	por_hnf_pmu_mpam_sel	RW	5.3.15.95 por_hnf_pmu_mpam_sel on page 820
16'h2010 + (8 × #[0, 1, ... 7])	por_hnf_pmu_mpam_pardid_mask0-7	RW	5.3.15.96 por_hnf_pmu_mpam_pardid_mask0-7 on page 822
16'h3C00 + (32 × #[0, 1, ... 63])	por_hnf_rn_cluster0-63_physid_reg0	RW	5.3.15.97 por_hnf_rn_cluster0-63_physid_reg0 on page 823
16'h3C00 + (32 × #[64, 65, ... 127])	por_hnf_rn_cluster64-127_physid_reg0	RW	5.3.15.98 por_hnf_rn_cluster64-127_physid_reg0 on page 826
16'h3C08 + (32 × #[0, 1, ... 127])	por_hnf_rn_cluster0-127_physid_reg1	RW	5.3.15.99 por_hnf_rn_cluster0-127_physid_reg1 on page 829
16'h3C10 + (32 × #[0, 1, ... 127])	por_hnf_rn_cluster0-127_physid_reg2	RW	5.3.15.100 por_hnf_rn_cluster0-127_physid_reg2 on page 831
16'h3C18 + (32 × #[0, 1, ... 127])	por_hnf_rn_cluster0-127_physid_reg3	RW	5.3.15.101 por_hnf_rn_cluster0-127_physid_reg3 on page 834

5.3 Register descriptions

This section contains register descriptions.

5.3.1 RN-D register descriptions

This section lists the RN-D registers.

5.3.1.1 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

Figure 5-1: por_rnd_node_info

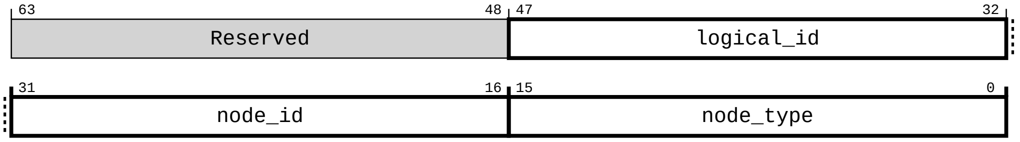


Table 5-17: por_rnd_node_info attributes

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-650 node type identifier	RO	16'h000D

5.3.1.2 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

Figure 5-2: por_rnd_child_info

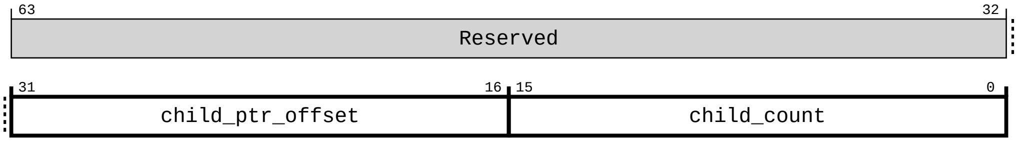


Table 5-18: por_rnd_child_info attributes

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. This value is used in the discovery process.	RO	16'h0

5.3.1.3 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

Figure 5-3: por_rnd_secure_register_groups_override

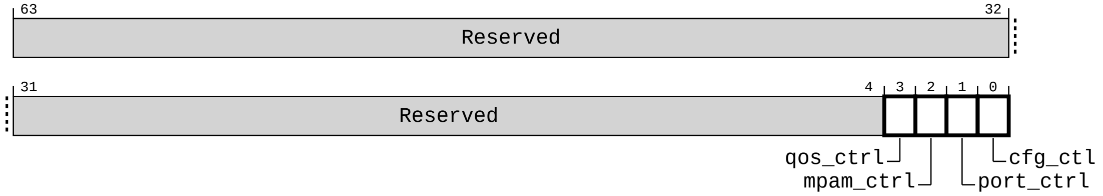


Table 5-19: por_rnd_secure_register_groups_override attributes

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

5.3.1.4 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

Figure 5-4: por_rnd_unit_info

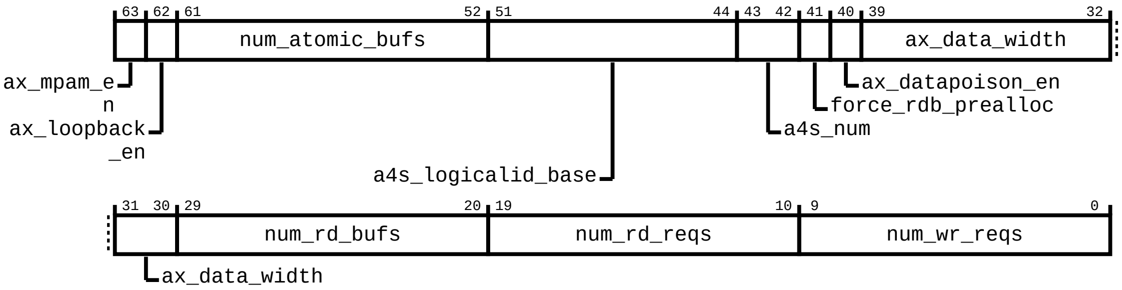


Table 5-20: por_rnd_unit_info attributes

Bits	Name	Description	Type	Reset
[63]	ax_mpam_en	MPAM enable on AXI/ACE-Lite interface: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[62]	ax_loopback_en	LoopBack enable on AXI/ACE-Lite interface: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[61:52]	num_atomic_bufs	Number of atomic data buffers	RO	Configuration dependent
[51:44]	a4s_logicalid_base	AXI4-Stream interfaces logical ID base	RO	Configuration dependent
[43:42]	a4s_num	Number of AXI4-Stream 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 AXI/ACE-Lite 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

5.3.1.5 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

Figure 5-5: por_rnd_unit_info2

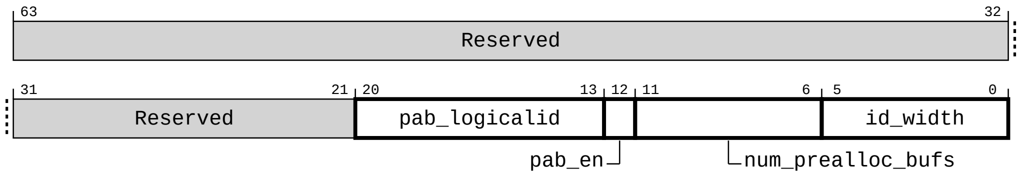


Table 5-21: por_rnd_unit_info2 attributes

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[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 preallocated read data buffers	RO	Configuration dependent
[5:0]	id_width	AXI ID width for ACE-Lite slave ports	RO	Configuration dependent

5.3.1.6 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

Figure 5-6: por_rnd_cfg_ctl

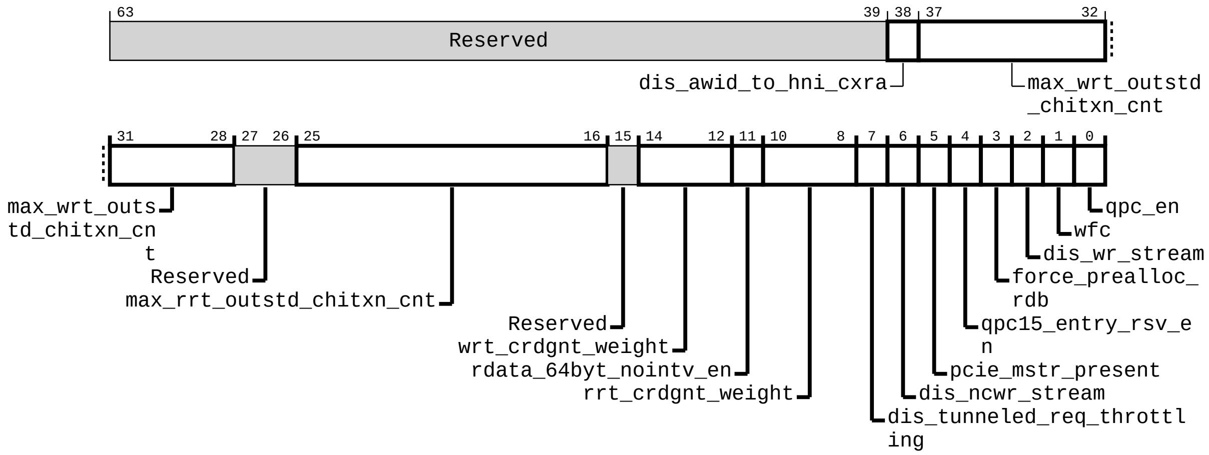


Table 5-22: por_rnd_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:39]	Reserved	Reserved	RO	-
[38]	dis_awid_to_hni_cxra	If set, disables compressed AWID to HN-I and CXRA, also disables compressed AWID-based ordering. Set this bit if unique-ID write performance is required.	RW	1'b0
[37:28]	max_wrt_outstd_chitxn_cnt	Maximum number of outstanding writes that are allowed on CHI-side	RW	Configuration dependent
[27:26]	Reserved	Reserved	RO	-
[25:16]	max_rrt_outstd_chitxn_cnt	Maximum number of outstanding reads that are allowed on CHI-side	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[15]	Reserved	Reserved	RO	-
[14:12]	wrt_crdgnt_weight	Determines weight of credit grant that is allocated to retried writes in presence of pending retried reads	RW	3'b001
[11]	rdata_64byt_nointv_en	If set, enables no interleaving property on normal memory read data within 64B granule	RW	1'b1
[10:8]	rrt_crdgnt_weight	Determines weight of credit grant that is 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_ncwr_stream	If set, disables streaming of ordered non-cacheable writes	RW	1'b0
[5]	pcie_mstr_present	Indicates PCIe master is present. This bit must be set if a PCIe master is present upstream of RN-I or RN-D.	RW	1'b0
[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	If set, all reads from the RN-D are sent with a preallocated read data buffer	RW	Configuration dependent
[2]	dis_wr_stream	If set, disables streaming of ordered writes	RW	1'b0
[1]	wfc	If set, enables waiting for completion (COMP) before dispatching dependent transaction (TXN)	RW	1'b0
[0]	qpc_en	If set, enables QoS Priority Class (QPC)-based scheduling using two QoS QPCs (QoS15 and non-QoS15)	RW	1'b1

5.3.1.7 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

Figure 5-7: por_rnd_aux_ctl

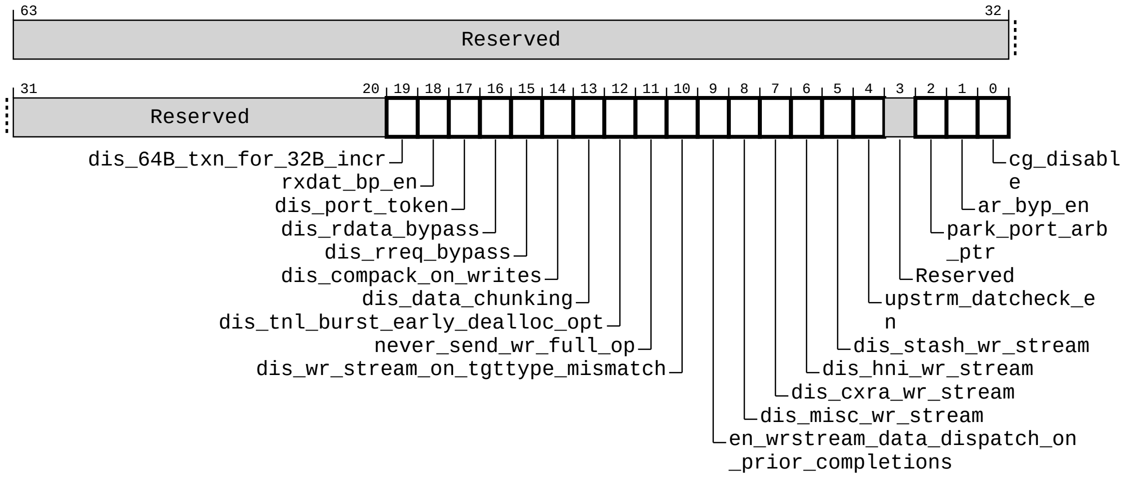


Table 5-23: por_rnd_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:20]	Reserved	Reserved	RO	-
[19]	dis_64B_txn_for_32B_incr	If set, disables the 2nd, 32B fake write for a 32B INCR burst (rh-2512)	RW	1'b0
[18]	rxdat_bp_en	If set, back pressures the RXDAT interface when read data buffers are not available	RW	1'b0
[17]	dis_port_token	If set, disables per port reservation in the tracker (read and write)	RW	1'b1
[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 tunneled writes for intermediate transactions of a burst	RW	1'b0
[11]	never_send_wr_full_op	If set, RN-I never sends write FULL operations. All write operations will be of PTL type	RW	1'b0
[10]	dis_wr_stream_on_tgttype_mismatch	If set, serializes first write when moving from one target type 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

Bits	Name	Description	Type	Reset
[8]	dis_misc_wr_stream	If set, disables streaming of ordered writes with following attributes: Device memory or EWA=0 or Excl=1	RW	1'b0
[7]	dis_cxra_wr_stream	If set, disables streaming of ordered writes to CXRA	RW	1'b0
[6]	dis_hni_wr_stream	If set, disables streaming of ordered writes to HN-I	RW	1'b0
[5]	dis_stash_wr_stream	If set, disables streaming of ordered WrUniqStash	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	If set, disables clock gating	RW	1'b0

5.3.1.8 por_rnd_s0-2_port_control

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls port S#{index} AXI/ACE slave interface settings.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA10 + (8 × #[0, 1, 2])

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

Figure 5-8: por_rnd_s0-2_port_control

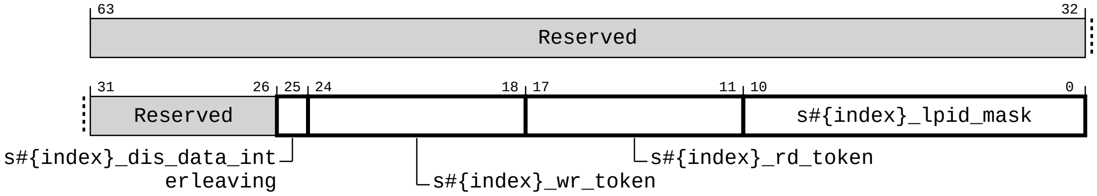


Table 5-24: por_rnd_s0-2_port_control attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	s#{index}_dis_data_interleaving	If set, disables read data interleaving on RDATAS#{index} channel. This setting only applies to RDATA generated as a response to requests on the AR channel. This setting does not apply to RDATA generated as a response to atomic requests on the AW channel. In other words, RDATA of an atomic operation on the AW channel can interleave with RDATA of an AR channel request.	RW	1'b0
[24:18]	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 channel.	RW	6'b00_0000
[17:11]	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 channel.	RW	6'b00_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

5.3.1.9 por_rnd_s0-2_mpam_control

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls port S#{index} AXI/ACE slave interface MPAM override values

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA28 + (8 × #[0, 1, 2])

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

Figure 5-9: por_rnd_s0-2_mpam_control

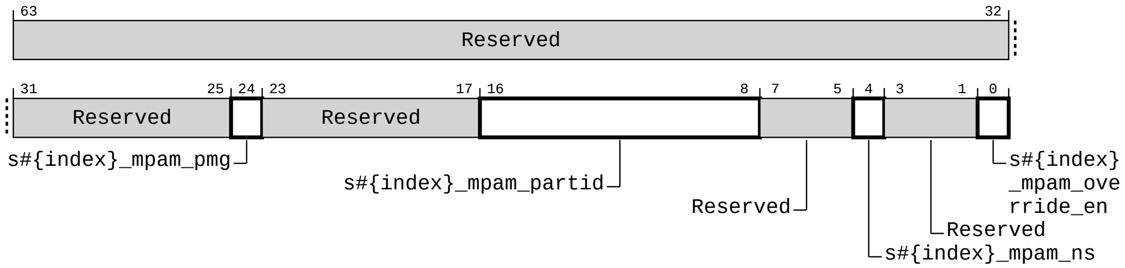


Table 5-25: por_rnd_s0-2_mpam_control attributes

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. If 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, the MPAM override value is always used, regardless of the value of this bit.	RW	1'b0

5.3.1.10 por_rnd_s0-2_qos_control

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls QoS settings for port S#{index} AXI/ACE slave interface.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA80 + (32 × #[0, 1, 2])

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

Figure 5-10: por_rnd_s0-2_qos_control

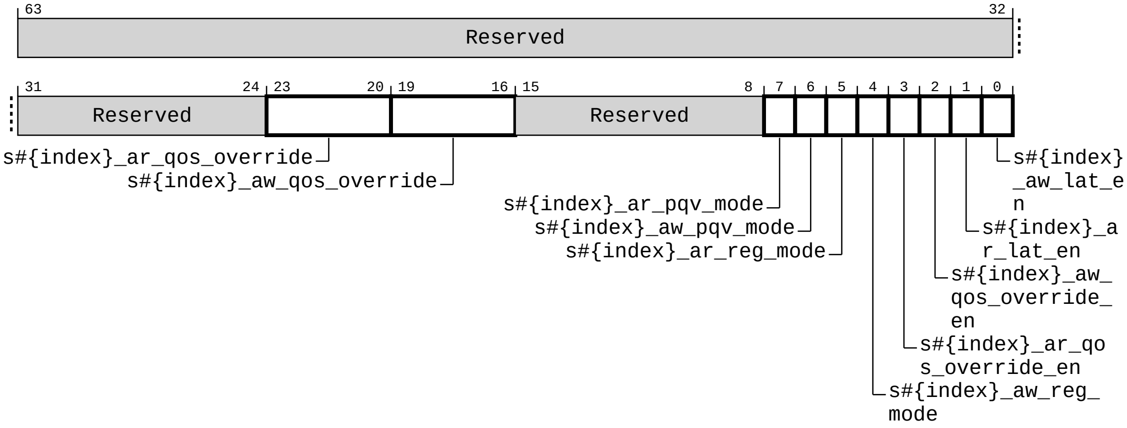


Table 5-26: por_rnd_s0-2_qos_control attributes

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	-

Bits	Name	Description	Type	Reset
[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 master is idle. 1'b1: Quiesce high mode. QoS value tends to the maximum value when the master 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 master is idle. 1'b1: Quiesce high mode. QoS value tends to the maximum value when the master is idle.	RW	1'b0
[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. If 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. If set, allows QoS value on inbound AW transactions to be overridden.	RW	1'b0
[1]	s#{index}_ar_lat_en	If set, enables port S#{index} AR QoS regulation	RW	1'b0
[0]	s#{index}_aw_lat_en	If set, enables port S#{index} AW QoS regulation	RW	1'b0

5.3.1.11 por_rnd_s0-2_qos_lat_tgt

There are 3 iterations of this register, parameterized by the index 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 + (32 × #[0, 1, 2])

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

Figure 5-11: por_rnd_s0-2_qos_lat_tgt

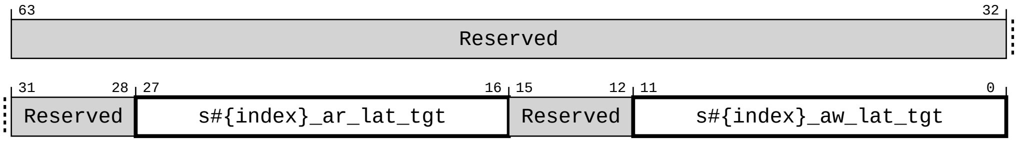


Table 5-27: por_rnd_s0-2_qos_lat_tgt attributes

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

5.3.1.12 por_rnd_s0-2_qos_lat_scale

There are 3 iterations of this register, parameterized by the index 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 + (32 × #[0, 1, 2])

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

Figure 5-12: por_rnd_s0-2_qos_lat_scale

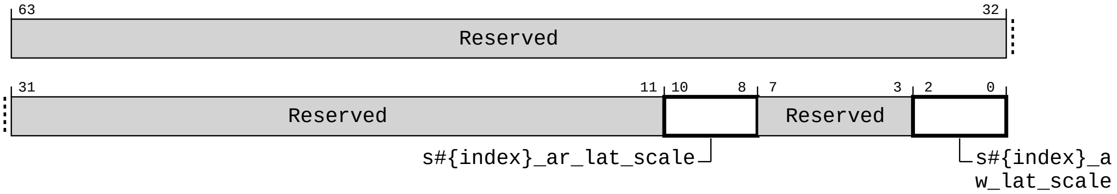


Table 5-28: por_rnd_s0-2_qos_lat_scale attributes

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	-

Bits	Name	Description	Type	Reset
[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

5.3.1.13 por_rnd_s0-2_qos_lat_range

There are 3 iterations of this register, parameterized by the index 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 + (32 × #[0, 1, 2])

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

Figure 5-13: por_rnd_s0-2_qos_lat_range

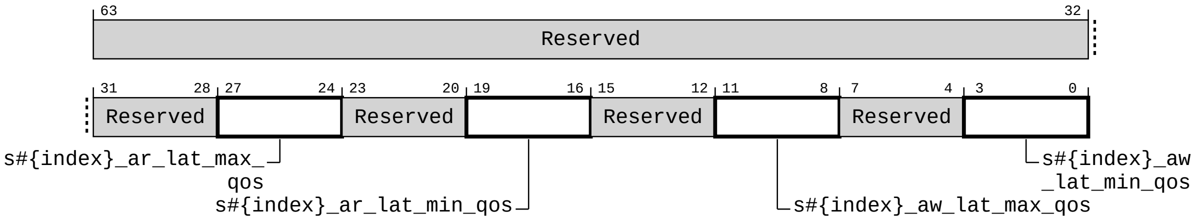


Table 5-29: por_rnd_s0-2_qos_lat_range attributes

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

5.3.1.14 por_rnd_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-14: por_rnd_pmu_event_sel

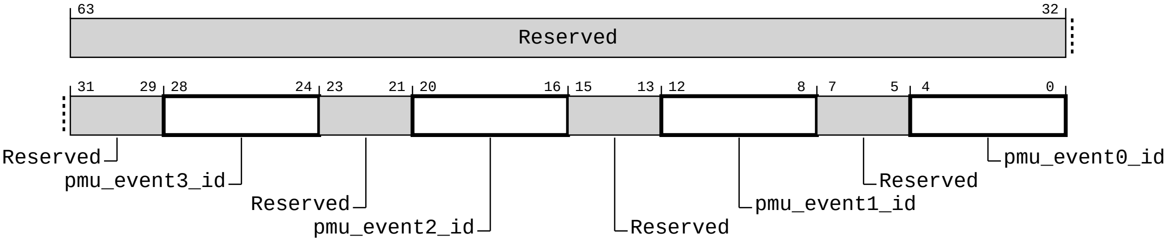


Table 5-30: por_rnd_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:24]	pmu_event3_id	RN-D PMU event 3 ID. See pmu_event0_id for encodings.	RW	5'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	pmu_event2_id	RN-D PMU event 2 ID. See pmu_event0_id for encodings.	RW	5'b0
[15:13]	Reserved	Reserved	RO	-
[12:8]	pmu_event1_id	RN-D PMU event 1 ID. See pmu_event0_id for encodings.	RW	5'b0
[7:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:0]	pmu_event0_id	RN-D PMU event 0 ID 5'h00: No event 5'h01: Port S0 RDataBeats 5'h02: Port S1 RDataBeats 5'h03: Port S2 RDataBeats 5'h04: RXDAT flits received 5'h05: TXDAT flits sent 5'h06: Total TXREQ flits sent 5'h07: Retried TXREQ flits sent 5'h08: RRT occupancy count overflow_slice0 5'h09: WRT occupancy count overflow 5'h0A: Replayed TXREQ flits 5'h0B: WriteCancel sent 5'h0C: Port S0 WDataBeats 5'h0D: Port S1 WDataBeats 5'h0E: Port S2 WDataBeats 5'h0F: RRT allocation 5'h10: WRT allocation 5'h11: PADB occupancy count overflow 5'h12: RPDB occupancy count overflow 5'h13: RRT occupancy count overflow_slice1 5'h14: RRT occupancy count overflow_slice2 5'h15: RRT occupancy count overflow_slice3 5'h16: WRT request throttled	RW	5'b0

5.3.1.15 por_rnd_syscoreq_ctl

Functions as the RN-D DVM domain control register. Provides a software alternative to hardware SYSCOREQ, SYSCOACK handshake. Works with the por_rnd_syscoack_status register.

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

Figure 5-15: por_rnd_syscoreq_ctl

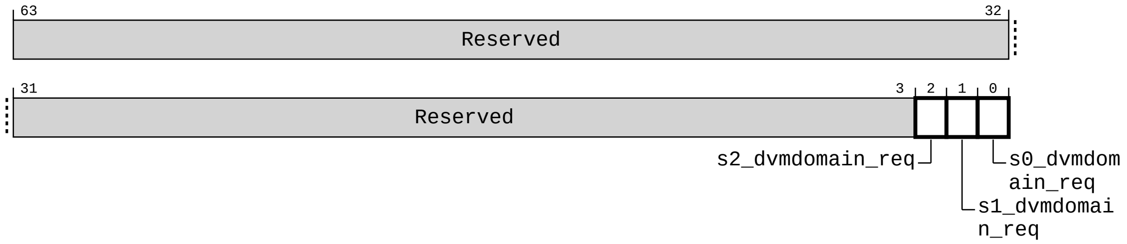


Table 5-31: por_rnd_syscoreq_ctl attributes

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

5.3.1.16 por_rnd_syscoack_status

Functions as the RN-D DVM domain status register. Provides a software alternative to hardware SYSCOREQ, SYSCOACK handshake. Works with the por_rnd_syscoreq_ctl register.

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

Figure 5-16: por_rnd_syscoack_status

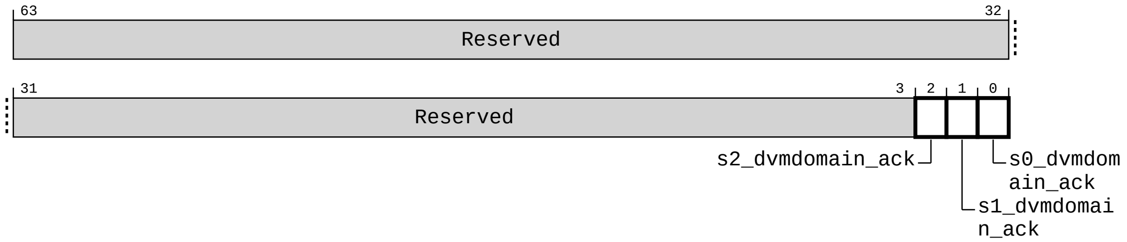


Table 5-32: por_rnd_syscoack_status attributes

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

5.3.2 CXRA register descriptions

This section lists the CXRA registers.

5.3.2.1 por_cxg_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

Figure 5-17: por_cxg_ra_node_info

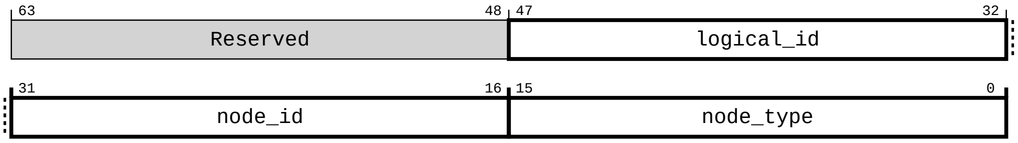


Table 5-33: por_cxg_ra_node_info attributes

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-650 node type identifier	RO	16'h0100

5.3.2.2 por_cxg_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

Figure 5-18: por_cxg_ra_child_info

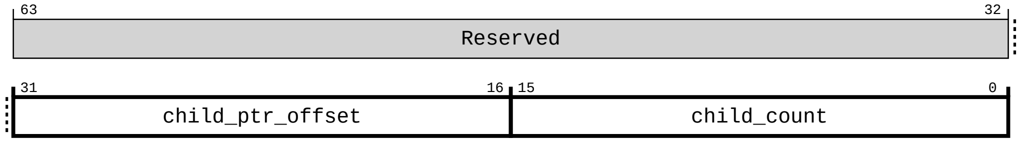


Table 5-34: por_cxg_ra_child_info attributes

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. This value is used in the discovery process.	RO	16'h0

5.3.2.3 por_cxg_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

Figure 5-19: por_cxg_ra_secure_register_groups_override

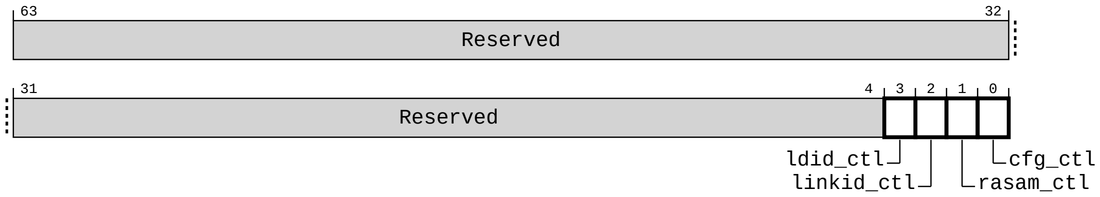


Table 5-35: por_cxg_ra_secure_register_groups_override attributes

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

5.3.2.4 por_cxg_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

Figure 5-20: por_cxg_ra_unit_info

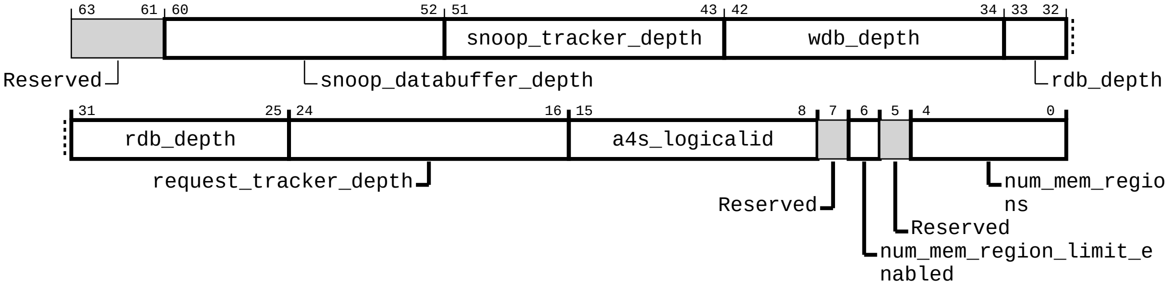


Table 5-36: por_cxg_ra_unit_info attributes

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:52]	snoop_databuffer_depth	Depth of snoop data buffer. The depth is the number of permitted outstanding SNP requests on CHI.	RO	Configuration dependent
[51:43]	snoop_tracker_depth	Depth of snoop tracker. The depth is the number of permitted outstanding SNP requests on CCIX.	RO	Configuration dependent
[42:34]	wdb_depth	Depth of write data buffer	RO	Configuration dependent
[33:25]	rdb_depth	Depth of read data buffer	RO	Configuration dependent
[24:16]	request_tracker_depth	Depth of request tracker. The depth is the number of permitted outstanding memory requests on CCIX.	RO	Configuration dependent
[15:8]	a4s_logicalid	AXI4-Stream interface logical ID	RO	Configuration dependent
[7]	Reserved	Reserved	RO	-
[6]	num_mem_region_limit_enabled	Memory region limiting enabled	RO	Configuration dependent
[5]	Reserved	Reserved	RO	-
[4:0]	num_mem_regions	Number of memory regions supported	RO	Configuration dependent

5.3.2.5 por_cxg_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_cxg_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

Figure 5-21: por_cxg_ra_cfg_ctl

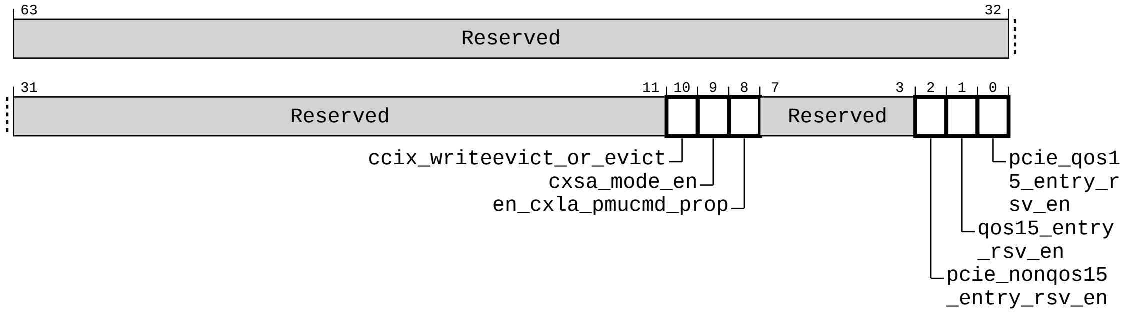


Table 5-37: por_cxg_ra_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	ccix_writeevict_or_evict	Is set, downgrades WriteEvict to Evict: 1'b1: Evict is sent instead of WriteEvict 1'b0: WriteEvict is sent	RW	1'b0
[9]	cxsa_mode_en	If set, enables CCIX Slave Agent (CXSA) mode. In this mode, CCIX Requesting Agent (CXRA) functions as a CXSA. 1'b1: CXSA 1'b0: CXRA	RW	1'b0
[8]	en_cxla_pmucmd_prop	If set, enables the propagation of PMU commands to CXLA NOTE: By default, CXLA PMU command propagation is disabled.	RW	1'b0

Bits	Name	Description	Type	Reset
[7:3]	Reserved	Reserved	RO	-
[2]	pcie_nonqos15_entry_rsv_en	Enables entry reservation for non-QoS15 traffic from PCIe RN-Is and RN-Ds: 1'b1: Reserves tracker entry for non-QoS15 requests from PCIe RN-Is and RN-Ds 1'b0: Does not reserve tracker entry for non-QoS15 requests from PCIe RN-Is and RN-Ds	RW	Configuration dependent
[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-Is and RN-Ds: 1'b1: Reserves tracker entry for QoS15 requests from PCIe RN-Is and RN-Ds 1'b0: Does not reserve tracker entry for QoS15 requests from PCIe RN-Is and RN-Ds	RW	Configuration dependent

5.3.2.6 por_cxg_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

Figure 5-22: por_cxg_ra_aux_ctl

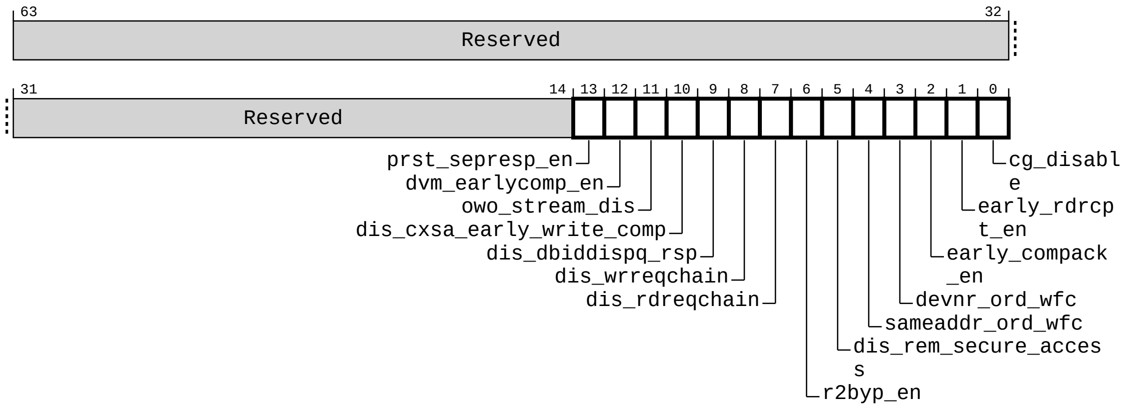


Table 5-38: por_cxg_ra_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13]	prst_sepresp_en	If set, enables separate persist response on CCIX for Persistent Cache Maintenance Operations (PCMO2). NOTE: This bit is applicable only in SMP mode.	RW	1'b1
[12]	dvm_earlycomp_en	If set, enables early DVMOp completion responses from RA	RW	1'b1
[11]	owo_stream_dis	If set, disables CompAck dependency to dispatch an ordered PCIe write	RW	1'b0
[10]	dis_cxsa_early_write_comp	If set, disables early write completions in CCIX Slave Agent (CXSA) mode	RW	1'b0
[9]	dis_dbiddispq_rsp	If set, disables the dispatch of DBID responses from a separate DispatchQ	RW	1'b0
[8]	dis_wrreqchain	If set, disables chaining of write requests	RW	1'b0
[7]	dis_rdreqchain	If set, disables chaining of read and dataless requests	RW	1'b0
[6]	r2byp_en	If set, enables request bypass. Applies to read and dataless requests only. NOTE: If set, this bit affects the capability to chain a request on the TX side.	RW	1'b0
[5]	dis_rem_secure_access	If set, treats all the incoming snoops as Non-secure and forces the NS bit to 1	RW	1'b0
[4]	sameaddr_ord_wfc	If set, enables waiting for completion (COMP) before dispatching next same address-dependent transaction (TXN)	RW	1'b0
[3]	devnr_ord_wfc	If 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
[1]	early_rdrcpt_en	Early ReadReceipt enable. Enables sending early ReadReceipt for ordered read requests.	RW	1'b1
[0]	cg_disable	If set, disables clock gating	RW	1'b0

5.3.2.7 por_cxg_ra_cbusy_limit_ctl

Completer Busy (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

Figure 5-23: por_cxg_ra_cbusy_limit_ctl

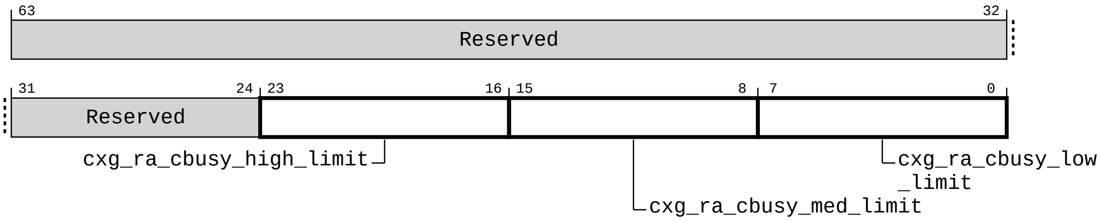


Table 5-39: por_cxg_ra_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	cxg_ra_cbusy_high_limit	RHT limit for CBusy high activity level	RW	Configuration dependent
[15:8]	cxg_ra_cbusy_med_limit	RHT limit for CBusy medium activity level	RW	Configuration dependent
[7:0]	cxg_ra_cbusy_low_limit	RHT limit for CBusy low activity level	RW	Configuration dependent

5.3.2.8 por_cxg_ra_sam_addr_region_reg0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Configures address region #{i} for the RA SAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC00 + (8 × #[0, 1, ... 7])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.rasam_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-24: por_cxg_ra_sam_addr_region_reg0-7

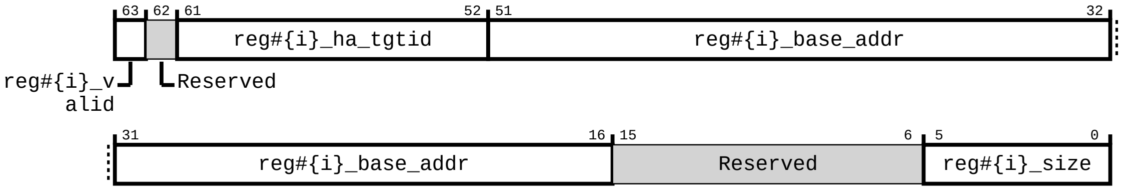


Table 5-40: por_cxg_ra_sam_addr_region_reg0-7 attributes

Bits	Name	Description	Type	Reset
[63]	reg#{i}_valid	Specifies if the memory region is valid	RW	1'b0
[62]	Reserved	Reserved	RO	-
[61:52]	reg#{i}_ha_tgtid	Specifies the target Home Agent ID (HAID) for the memory region	RW	10'b0
[51:16]	reg#{i}_base_addr	Specifies the 2^n-aligned base address for the memory region	RW	36'h0
[15:6]	Reserved	Reserved	RO	-
[5:0]	reg#{i}_size	Specifies the size of the memory region	RW	1'b0

5.3.2.9 por_cxg_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_cxg_ra_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-25: por_cxg_ra_agentid_to_linkid_val

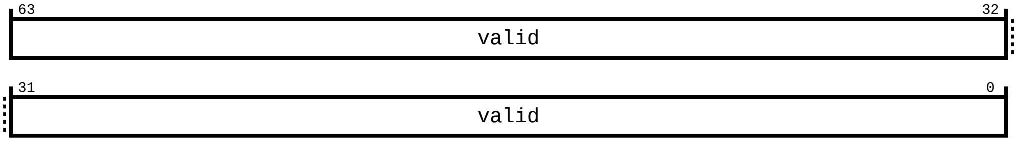


Table 5-41: por_cxg_ra_agentid_to_linkid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Link ID is valid. The bit number corresponds to the logical Agent ID number (from 0-63)	RW	63'h0

5.3.2.10 por_cxg_ra_agentid_to_linkid_reg0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Specifies the mapping of Agent ID to Link ID for Agent IDs #{i*8} to #{i*8+7}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD10 + (8 × #[0, 1, ... 7])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-26: por_cxg_ra_agentid_to_linkid_reg0-7

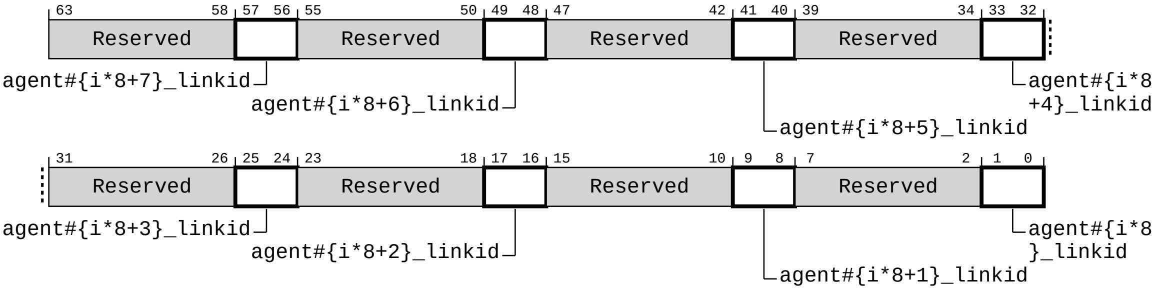


Table 5-42: por_cxg_ra_agentid_to_linkid_reg0-7 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent#{i*8+7}_linkid	Specifies the Link ID for Agent ID #{i*8+7}	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent#{i*8+6}_linkid	Specifies the Link ID for Agent ID #{i*8+6}	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent#{i*8+5}_linkid	Specifies the Link ID for Agent ID #{i*8+5}	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent#{i*8+4}_linkid	Specifies the Link ID for Agent ID #{i*8+4}	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent#{i*8+3}_linkid	Specifies the Link ID for Agent ID #{i*8+3}	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent#{i*8+2}_linkid	Specifies the Link ID for Agent ID #{i*8+2}	RW	2'h0

Bits	Name	Description	Type	Reset
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent#{i*8+1}_linkid	Specifies the Link ID for Agent ID #{i*8+1}	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent#{i*8}_linkid	Specifies the Link ID for Agent ID #{i*8}	RW	2'h0

5.3.2.11 por_cxg_ra_rni_ldid_to_exp_raid_reg0-9

There are 10 iterations of this register, parameterized by the index from 0 to 9. Specifies the mapping of the RN-I LDID to expanded RAID, for LDIDs #{i*4} to #{i*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE00 + (8 × #[0, 1, ... 9])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-27: por_cxg_ra_rni_ldid_to_exp_raid_reg0-9

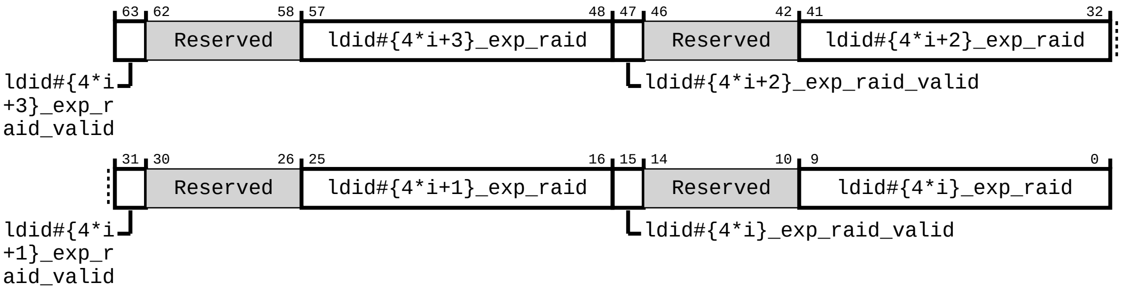


Table 5-43: por_cxg_ra_rni_ldid_to_exp_raid_reg0-9 attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{4*i+3}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{4*i+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-
[57:48]	ldid#{4*i+3}_exp_raid	Specifies the expanded RAID for LDID #{4*i+3}	RW	10'h0
[47]	ldid#{4*i+2}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{4*i+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{4*i+2}_exp_raid	Specifies the expanded RAID for LDID #{4*i+2}	RW	10'h0
[31]	ldid#{4*i+1}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{4*i+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{4*i+1}_exp_raid	Specifies the expanded RAID for LDID #{4*i+1}	RW	10'h0
[15]	ldid#{4*i}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{4*i} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{4*i}_exp_raid	Specifies the expanded RAID for LDID #{4*i}	RW	10'h0

5.3.2.12 por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9

There are 10 iterations of this register, parameterized by the index from 0 to 9. Specifies the mapping of the RN-D LDID to expanded RAID, for LDIDs #{i*4} to #{i*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF00 + (8 × #[0, 1, ... 9])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-28: por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9

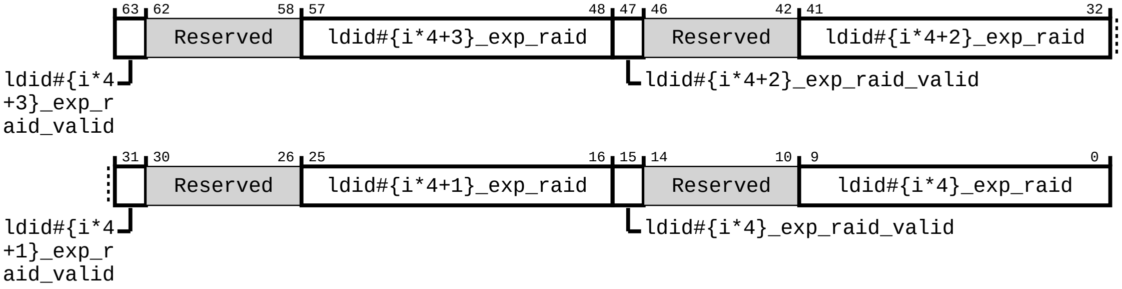


Table 5-44: por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9 attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{i*4+3}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{i*4+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-
[57:48]	ldid#{i*4+3}_exp_raid	Specifies the expanded RAID for LDID #{i*4+3}	RW	10'h0
[47]	ldid#{i*4+2}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{i*4+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{i*4+2}_exp_raid	Specifies the expanded RAID for LDID #{i*4+2}	RW	10'h0
[31]	ldid#{i*4+1}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{i*4+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{i*4+1}_exp_raid	Specifies the expanded RAID for LDID #{i*4+1}	RW	10'h0
[15]	ldid#{i*4}_exp_raid_valid	Specifies whether the expanded RAID for LDID#{i*4} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{i*4}_exp_raid	Specifies the expanded RAID for LDID #{i*4}	RW	10'h0

5.3.2.13 por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127

There are 128 iterations of this register, parameterized by the index from 0 to 127. Specifies the mapping of the RN-F LDID to expanded RAID, for LDIDs #{i*4} to #{i*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1000 + (8 × #[0, 1, ... 127])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-29: por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127

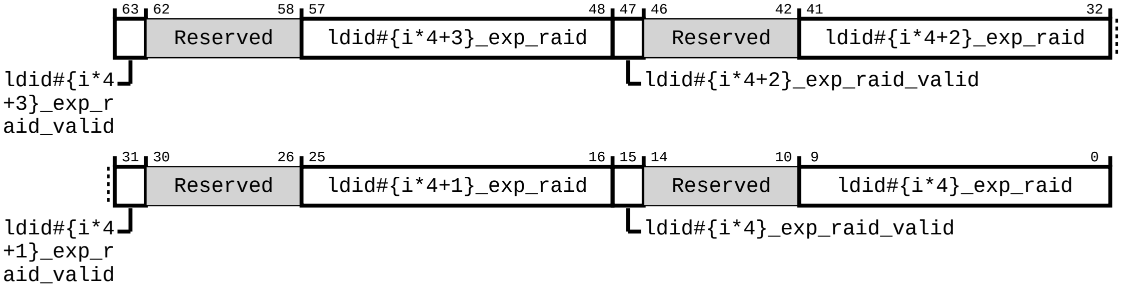


Table 5-45: por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127 attributes

Bits	Name	Description	Type	Reset
[63]	ldid#{i*4+3}_exp_raid_valid	Specifies whether the lookup table entry for default LDID#{i*4+3} is valid;	RW	1'h0
[62:58]	Reserved	Reserved	RO	-
[57:48]	ldid#{i*4+3}_exp_raid	Specifies the expanded RAID for LDID #{i*4+3}	RW	10'h0
[47]	ldid#{i*4+2}_exp_raid_valid	Specifies whether the lookup table entry for default LDID#{i*4+2} is valid;	RW	1'h0
[46:42]	Reserved	Reserved	RO	-
[41:32]	ldid#{i*4+2}_exp_raid	Specifies the expanded RAID for LDID #{i*4+2}	RW	10'h0
[31]	ldid#{i*4+1}_exp_raid_valid	Specifies whether the lookup table entry for default LDID#{i*4+1} is valid;	RW	1'h0
[30:26]	Reserved	Reserved	RO	-
[25:16]	ldid#{i*4+1}_exp_raid	Specifies the expanded RAID for LDID #{i*4+1}	RW	10'h0
[15]	ldid#{i*4}_exp_raid_valid	Specifies whether the lookup table entry for default LDID#{i*4} is valid;	RW	1'h0
[14:10]	Reserved	Reserved	RO	-
[9:0]	ldid#{i*4}_exp_raid	Specifies the expanded RAID for LDID #{i*4}	RW	10'h0

5.3.2.14 por_cxg_ra_rnf_ldid_to_nodeid_reg0-127

There are 128 iterations of this register, parameterized by the index from 0 to 127. Specifies the mapping of the RN-F default LDID to CHI NodeID, for LDIDs #{i*4} to #{i*4+3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1400 + (8 × #[0, 1, ... 127])

Type

RO

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-30: por_cxg_ra_rnf_ldid_to_nodeid_reg0-127

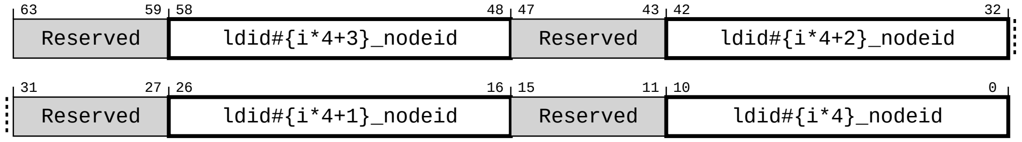


Table 5-46: por_cxg_ra_rnf_ldid_to_nodeid_reg0-127 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	ldid#{i*4+3}_nodeid	Specifies the CHI NodeID for LDID #{i*4+3}	RO	11'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	ldid#{i*4+2}_nodeid	Specifies the CHI NodeID for LDID #{i*4+2}	RO	11'h0
[31:27]	Reserved	Reserved	RO	-
[26:16]	ldid#{i*4+1}_nodeid	Specifies the CHI NodeID for LDID #{i*4+1}	RO	11'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	ldid#{i*4}_nodeid	Specifies the CHI NodeID for LDID #{i*4}	RO	11'h0

5.3.2.15 por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127

There are 128 iterations of this register, parameterized by the index from 0 to 127. Specifies the mapping of the RN-F override LDID, for default LDIDs #{i*4} to #{i*4+3}. This register is valid only if POR_MXP_RNF_CLUSTER_EN_PARAM = 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1800 + (8 × #[0, 1, ... 127])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ra_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-31: por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127

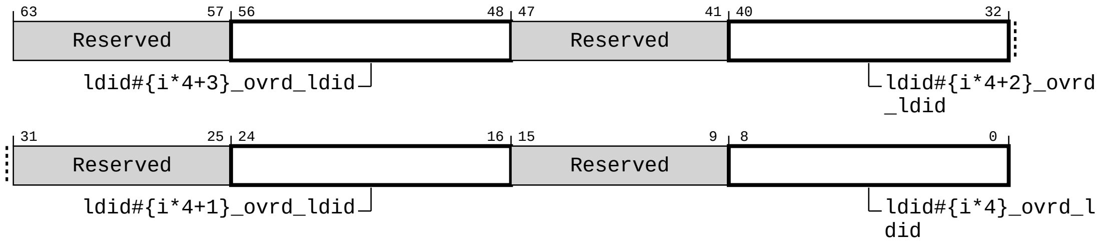


Table 5-47: por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127 attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ldid#{i*4+3}_ovrd_ldid	Specifies the overridden LDID for default LDID #{i*4+3}	RW	#{i*4+3}
[47:41]	Reserved	Reserved	RO	-
[40:32]	ldid#{i*4+2}_ovrd_ldid	Specifies the overridden LDID for default LDID #{i*4+2}	RW	{i*4+2}
[31:25]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[24:16]	ldid#{i*4+1}_ovrd_ldid	Specifies the overridden LDID for default LDID #{i*4+1}	RW	{i*4+1}
[15:9]	Reserved	Reserved	RO	-
[8:0]	ldid#{i*4}_ovrd_ldid	Specifies the overridden LDID for default LDID #{i*4}	RW	{i*4}

5.3.2.16 por_cxg_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

Figure 5-32: por_cxg_ra_pmu_event_sel

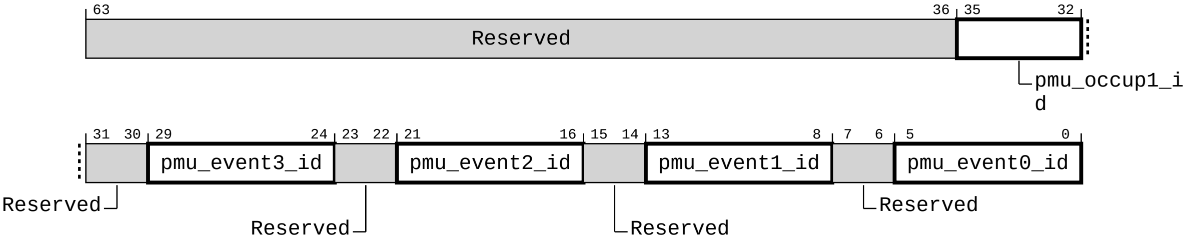


Table 5-48: por_cxg_ra_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	-
[35:32]	pmu_occup1_id	PMU occupancy event selector ID	RW	4'b0
[31:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	CXRA PMU event 3 ID. See pmu_event0_id for encodings.	RW	6'b0

Bits	Name	Description	Type	Reset
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	CXRA PMU event 2 ID. See pmu_event0_id for encodings.	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	CXRA PMU event 1 ID. See pmu_event0_id for encodings.	RW	6'b0
[7:6]	Reserved	Reserved	RO	-
[5:0]	pmu_event0_id	CXRA PMU event 0 ID: 6'h00: No event 6'h01: Request Tracker (RHT) occupancy count overflow 6'h02: Snoop Tracker (SHT) occupancy count overflow 6'h03: Read Data Buffer (RDB) occupancy count overflow 6'h04: Write Data Buffer (WDB) occupancy count overflow 6'h05: Snoop Sink Buffer (SSB) occupancy count overflow 6'h06: CCIX RX broadcast snoops 6'h07: CCIX TX request chain 6'h08: CCIX TX request chain average length 6'h09: CHI internal RSP stall 6'h0A: CHI internal DAT stall 6'h0B: CCIX REQ Protocol credit Link 0 stall 6'h0C: CCIX REQ Protocol credit Link 1 stall 6'h0D: CCIX REQ Protocol credit Link 2 stall 6'h0E: CCIX DAT Protocol credit Link 0 stall 6'h0F: CCIX DAT Protocol credit Link 1 stall 6'h10: CCIX DAT Protocol credit Link 2 stall 6'h11: CHI external RSP stall 6'h12: CHI external DAT stall 6'h13: CCIX MISC Protocol credit Link 0 stall 6'h14: CCIX MISC Protocol credit Link 1 stall 6'h15: CCIX MISC Protocol credit Link 2 stall	RW	6'b0

5.3.2.17 por_cxg_ra_cxprtcl_link0_ctl

Functions as the CXRA CCIX protocol link 0 control register. Works with the por_cxg_ra_cxprtcl_link0_status register.

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

Figure 5-33: por_cxg_ra_cxprtcl_link0_ctl

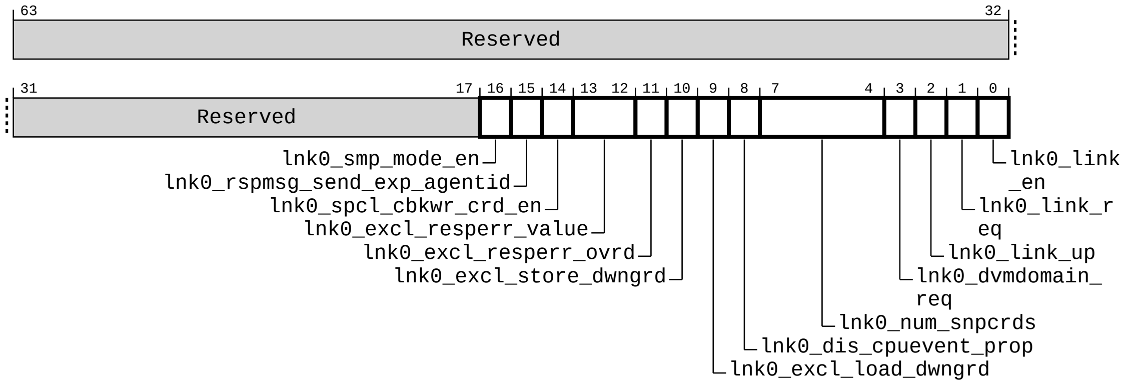


Table 5-49: por_cxg_ra_cxprtcl_link0_ctl attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	lnk0_smp_mode_en	If set, enables SMP mode for CCIX link 0.	RW	Configuration dependent
[15]	lnk0_rspmsg_send_exp_agentid	If set, sends expanded Agent ID on CCIX response messages for CCIX link 0	RW	1'b0

Bits	Name	Description	Type	Reset
[14]	lnk0_spcl_cbkwr_crd_en	If set, notifies RA to use special credits from HA to send CopyBack writes on CCIX link 0 NOTE: This field is only applicable if the link is programmed for SMP mode, in other words the SMP mode enable bit is set.	RW	1'b0
[13:12]	lnk0_excl_resperr_value	2-bit value to override RespErr field of an exclusive response. Applicable only if the lnk0_excl_resperr_ovrd bit is set. NOTE: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	2'b0
[11]	lnk0_excl_resperr_ovrd	If set, overrides the RespErr field of exclusive response with the lnk0_excl_resperr_value field NOTE: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[10]	lnk0_excl_store_dwngrd	If set, downgrades shareable exclusive store to shareable store when sending on CCIX link 0 Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[9]	lnk0_excl_load_dwngrd	If set, downgrades shareable exclusive load to shareable load when sending on CCIX link 0 Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[8]	lnk0_dis_cpuevent_prop	If set, disables the propagation of CPU events on CCIX link 0 NOTE: This field is only applicable when Symmetric Multiprocessor (SMP) mode is enabled.	RW	1'b0
[7:4]	lnk0_num_snpcrds	Controls the number of CCIX snoop credits that are 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

Bits	Name	Description	Type	Reset
[2]	lnk0_link_up	Link up status. Software writes this register bit to indicate link status after polling lnk<x>_link_ack and lnk<x>_link_down status in the remote agent: 1'b0: Link is not up. Software clears lnk<x>_link_up when lnk<x>_link_ack is clear and lnk<x>_link_down 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 lnk<x>_link_up is clear. 1'b1: Link is up. Software sets lnk<x>_link_up when lnk<x>_link_ack is set and lnk<x>_link_down is clear in both local and remote agents. The local agent starts sending local protocol credits to the remote agent.	RW	1'b0
[1]	lnk0_link_req	Request link up or link down. Software writes this register bit to request a link up or link down in the local agent: 1'b0: Link down request NOTE: The local agent does not return remote protocol credits yet, because the remote agent may still be in link up state. 1'b1: Link up request	RW	1'b0
[0]	lnk0_link_en	If set, enables CCIX link 0: 1'b0: Link is disabled 1'b1: Link is enabled	RW	1'b0

5.3.2.18 por_cxg_ra_cxprtcl_link0_status

Functions as the CXRA CCIX protocol link 0 status register. Works with the por_cxg_ra_cxprtcl_link0_ctl register.

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

Figure 5-34: por_cxg_ra_cxprtcl_link0_status

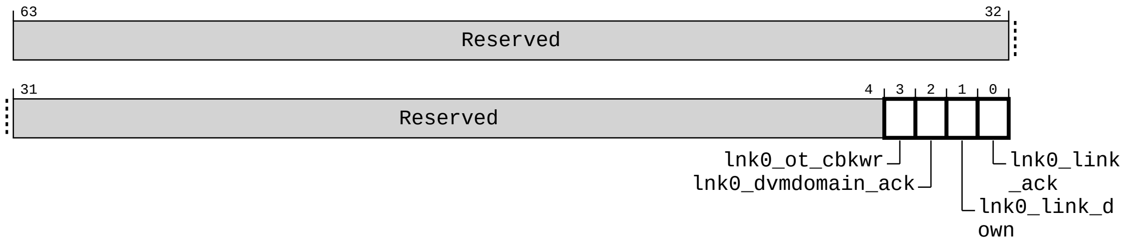


Table 5-50: por_cxg_ra_cxprtcl_link0_status attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	lnk0_ot_cbkwr	Provides status for outstanding CopyBack Write for CCIX link 0	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. The hardware updates this register bit to indicate link down status: 1'b0: Link is not down. The hardware clears lnk<x>_link_down when it receives a link up request. 1'b1: Link is down. The hardware sets lnk<x>_link_down 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 lnk<x>_link_up is clear.	RO	1'b1
[0]	lnk0_link_ack	Acknowledge link up and link down requests. The hardware updates this register bit to acknowledge the software link request: 1'b0: Link down acknowledged. The hardware clears lnk<x>_link_ack when it receives a link down request. The local agent stops granting protocol credits and starts returning protocol credits to the remote agent when lnk<x>_link_ack is clear. 1'b1: Link up acknowledged. The hardware sets lnk<x>_link_ack when the local agent is ready to start accepting protocol credits from the remote agent. Note: The local agent must clear lnk<x>_link_down before setting lnk<x>_link_ack.	RO	1'b0

5.3.2.19 por_cxg_ra_cxprtcl_link1_ctl

Functions as the CXRA CCIX protocol link 1 control register. Works with the por_cxg_ra_cxprtcl_link1_status register.

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

Figure 5-35: por_cxg_ra_cxprtcl_link1_ctl

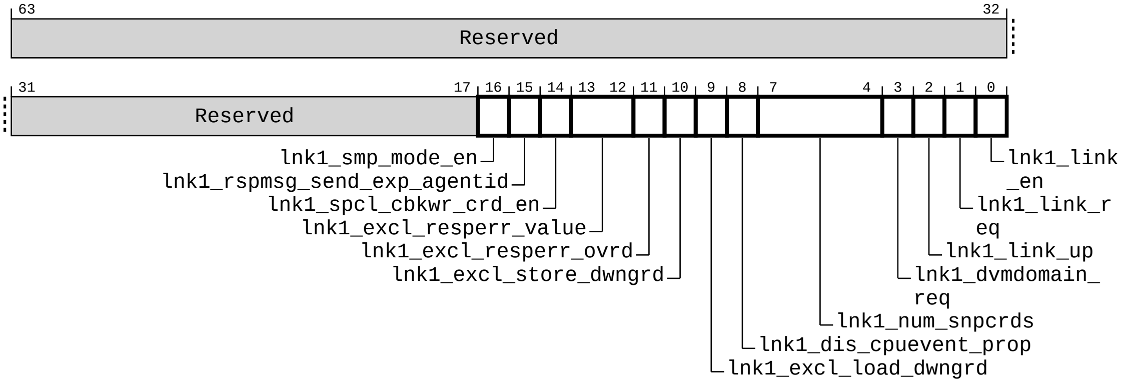


Table 5-51: por_cxg_ra_cxprtcl_link1_ctl attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	lnk1_smp_mode_en	If set, enables SMP mode for CCIX link 1	RW	Configuration dependent
[15]	lnk1_rspmsg_send_exp_agentid	If set sends expanded Agent ID on CCIX response messages for CCIX link 1	RW	1'b0
[14]	lnk1_spcl_cbkwr_crd_en	If set, notifies RA to use special credits from HA to send CopyBack writes on CCIX link 1 Note: This field is only applicable if the link is programmed for SMP mode, in other words the SMP mode enable bit is set.	RW	1'b0
[13:12]	lnk1_excl_resperr_value	2-bit value to override RespErr field of an exclusive response. Applicable only if the lnk1_excl_resperr_ovrd bit is set. NOTE: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	2'b0

Bits	Name	Description	Type	Reset
[11]	lnk1_excl_resperr_ovrd	If set, overrides the RespErr field of exclusive response with the lnk1_excl_resperr_value field Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[10]	lnk1_excl_store_dwngrd	If set, downgrades shareable exclusive store to shareable store when sending on CCIX link 1 Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[9]	lnk1_excl_load_dwngrd	If set, downgrades shareable exclusive load to shareable load when sending on CCIX link 1 NOTE: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[8]	lnk1_dis_cpuevent_prop	If set, disables the propagation of CPU events on CCIX link 1 Note: This field is only applicable when the SMP mode enable parameter is set.	RW	1'b0
[7:4]	lnk1_num_snpcrds	Controls the number of CCIX snoop credits that are 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 lnk<x>_link_ack and lnk<x>_link_down status in the remote agent: 1'b0: Link is not up. Software clears lnk<x>_link_up when lnk<x>_link_ack is clear and lnk<x>_link_down 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 lnk<x>_link_up is clear. 1'b1: Link is up. Software sets lnk<x>_link_up when lnk<x>_link_ack is set and lnk<x>_link_down is clear in both local and remote agents. The local agent starts sending local protocol credits to the remote agent.	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	lnk1_link_req	Request link up or link down. Software writes this register bit to request a link up or link down in the local agent: 1'b0: Link down request Note: The local agent does not return remote protocol credits yet, because the remote agent may still be in link up state. 1'b1: Link up request	RW	1'b0
[0]	lnk1_link_en	If set, enables CCIX link 1: 1'b0: Link is disabled 1'b1: Link is enabled	RW	1'b0

5.3.2.20 por_cxg_ra_cxprtcl_link1_status

Functions as the CXRA CCIX protocol link 1 status register. Works with the por_cxg_ra_cxprtcl_link1_ctl register.

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

Figure 5-36: por_cxg_ra_cxprtcl_link1_status

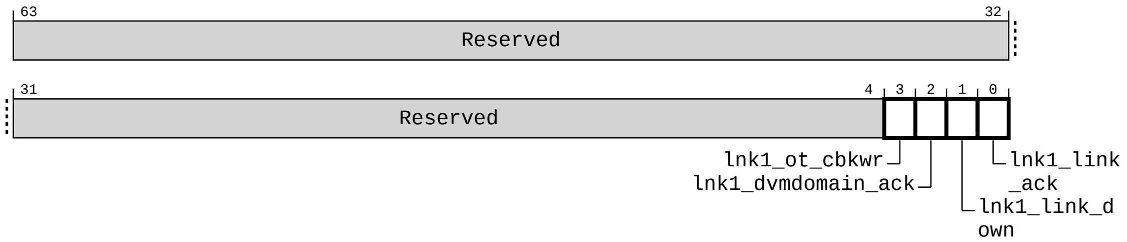


Table 5-52: por_cxg_ra_cxprtcl_link1_status attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[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. The hardware updates this register bit to indicate link down status: 1'b0: Link is not down. The hardware clears lnk<x>_link_down when it receives a link up request. 1'b1: Link is down. The hardware sets lnk<x>_link_down 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 lnk<x>_link_up is clear.	RO	1'b1
[0]	lnk1_link_ack	Acknowledge link up and link down requests. The hardware updates this register bit to acknowledge the software link request: 1'b0: Link down acknowledged. The hardware clears lnk<x>_link_ack when it receives a link down request. The local agent stops sending protocol credits to the remote agent when lnk<x>_link_ack is clear. 1'b1: Link up acknowledged. The hardware sets lnk<x>_link_ack when the local agent is ready to start accepting protocol credits from the remote agent. Note: The local agent must clear lnk<x>_link_down before setting lnk<x>_link_ack.	RO	1'b0

5.3.2.21 por_cxg_ra_cxprtcl_link2_ctl

Functions as the CXRA CCIX protocol link 2 control register. Works with the por_cxg_ra_cxprtcl_link2_status register.

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

Figure 5-37: por_cxg_ra_cxprtcl_link2_ctl

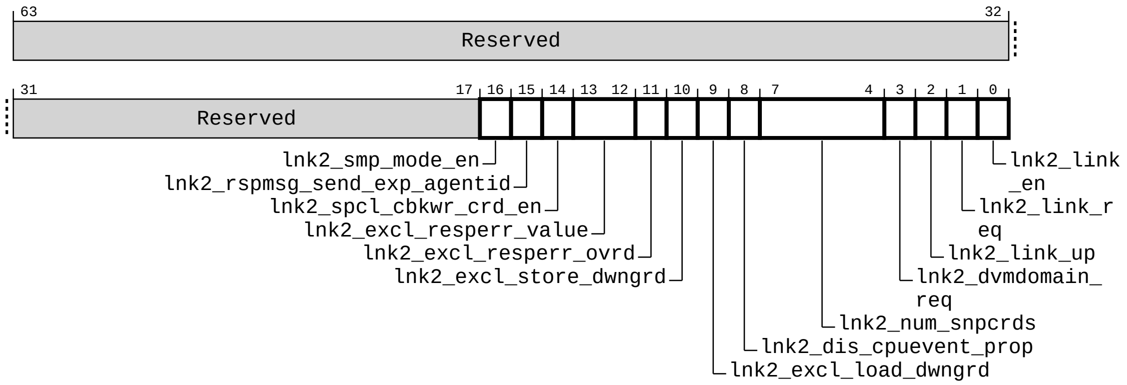


Table 5-53: por_cxg_ra_cxprtcl_link2_ctl attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	lnk2_smp_mode_en	If set, enables SMP mode for CCIX link 2	RW	Configuration dependent
[15]	lnk2_rspmsg_send_exp_agentid	If set, sends expanded Agent ID on CCIX response messages for CCIX link 2	RW	1'b0
[14]	lnk2_spcl_cbkwr_crd_en	If set, notifies RA to use special credits from HA to send CopyBack writes on CCIX link 2 Note: This field is only applicable if the link is programmed for SMP mode, in other words the SMP mode enable bit is set.	RW	1'b0
[13:12]	lnk2_excl_resperr_value	2-bit value to override RespErr field of an exclusive response. Applicable only if lnk2_excl_resperr_ovrd bit is set. NOTE: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	2'b0

Bits	Name	Description	Type	Reset
[11]	lnk2_excl_resperr_ovrd	If set, overrides the RespErr field of exclusive response with the lnk2_excl_resperr_value field Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[10]	lnk2_excl_store_dwngrd	If set, downgrades shareable exclusive store to shareable store when sending on CCIX link 2 Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[9]	lnk2_excl_load_dwngrd	If set, downgrades shareable exclusive load to shareable load when sending on CCIX link 2 Note: This field is only applicable when the SMP mode enable bit is clear, in other words the CXRA is in non-SMP mode. It must only be used if the corresponding link end pair does not support exclusive monitoring.	RW	1'b0
[8]	lnk2_dis_cpuevent_prop	If set, disables the propagation of CPU events on CCIX link 2 Note: This field is only applicable when the SMP mode enable parameter is set.	RW	1'b0
[7:4]	lnk2_num_snpcrds	Controls the number of CCIX snoop credits that are 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 lnk<x>_link_ack and lnk<x>_link_down status in the remote agent: 1'b0: Link is not up. Software clears lnk<x>_link_up when lnk<x>_link_ack is clear and lnk<x>_link_down 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 lnk<x>_link_up is clear. 1'b1: Link is up. Software sets lnk<x>_link_up when lnk<x>_link_ack is set and lnk<x>_link_down is clear in both local and remote agents. The local agent starts sending local protocol credits to the remote agent.	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	lnk2_link_req	Request link up or link down. Software writes this register bit to request a link up or link down in the local agent: 1'b0: Link down request NOTE: The local agent does not return remote protocol credits yet, because the remote agent may still be in link up state. 1'b1: link up request	RW	1'b0
[0]	lnk2_link_en	If set, enables CCIX link 2: 1'b0: Link is disabled 1'b1: Link is enabled	RW	1'b0

5.3.2.22 por_cxg_ra_cxprtcl_link2_status

Functions as the CXRA CCIX protocol link 2 status register. Works with the por_cxg_ra_cxprtcl_link2_ctl register.

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

Figure 5-38: por_cxg_ra_cxprtcl_link2_status

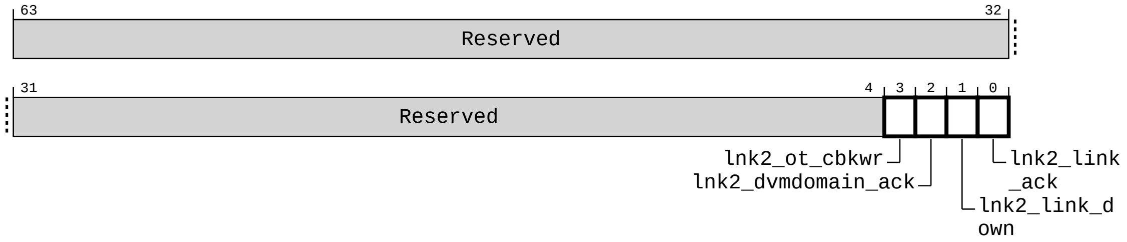


Table 5-54: por_cxg_ra_cxprtcl_link2_status attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[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. The hardware updates this register bit to indicate link down status: 1'b0: Link is not down. The hardware clears lnk<x>_link_down when it receives a link up request. 1'b1: Link is down. The hardware sets lnk<x>_link_down 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 lnk<x>_link_up is clear.	RO	1'b1
[0]	lnk2_link_ack	Acknowledge link up and link down requests. The hardware updates this register bit to acknowledge the software link request: 1'b0: Link down acknowledged. The hardware clears lnk<x>_link_ack when it receives a link down request. The local agent stops sending protocol credits to the remote agent when lnk<x>_link_ack is clear. 1'b1: Link up acknowledged. The hardware sets lnk<x>_link_ack when the local agent is ready to start accepting protocol credits from the remote agent. Note: The local agent must clear lnk<x>_link_down before setting lnk<x>_link_ack.	RO	1'b0

5.3.3 HN-I register descriptions

This section lists the HN-I registers.

5.3.3.1 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

Figure 5-39: por_hni_node_info

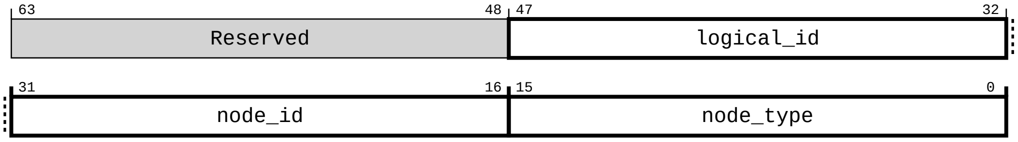


Table 5-55: por_hni_node_info attributes

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-650 node type identifier	RO	Configuration dependent

5.3.3.2 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

Figure 5-40: por_hni_child_info

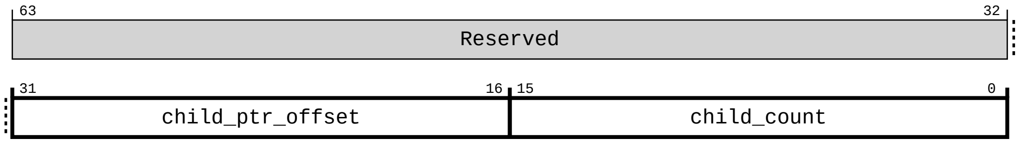


Table 5-56: por_hni_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.3.3 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

Figure 5-41: por_hni_secure_register_groups_override

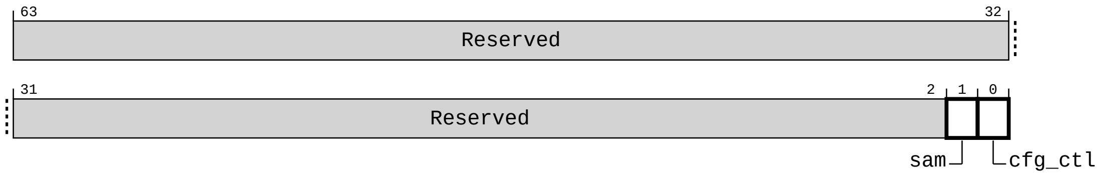


Table 5-57: por_hni_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:2]	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

5.3.3.4 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

Figure 5-42: por_hni_unit_info

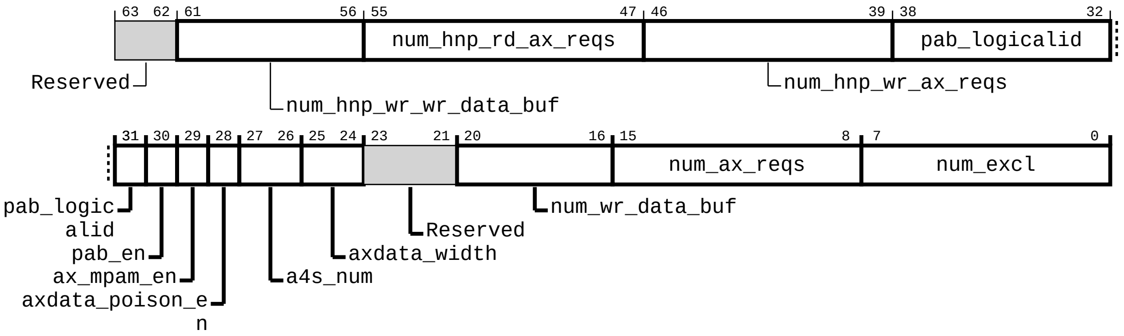


Table 5-58: por_hni_unit_info attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:56]	num_hnp_wr_wr_data_buf	Number of peer-to-peer write data buffers in HN-I. This field only applies to HN-P nodes.	RO	Configuration dependent
[55:47]	num_hnp_rd_ax_reqs	Maximum number of outstanding peer-to-peer read AXI/ACE-Lite requests. This field only applies to HN-P nodes.	RO	Configuration dependent
[46:39]	num_hnp_wr_ax_reqs	Maximum number of outstanding peer-to-peer write AXI/ACE-Lite requests. This field only applies to HN-P nodes.	RO	Configuration dependent
[38:31]	pab_logicalid	PUB AUB bridge Logical ID	RO	Configuration dependent
[30]	pab_en	PUB AUB bridge enable: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[29]	ax_mpam_en	MPAM enable on AXI/ACE-Lite interface: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[28]	axdata_poison_en	Data poison support on AXI/ACE-Lite interface: 1'b0: Not supported 1'b1: Supported	RO	Configuration dependent
[27:26]	a4s_num	Number of AXI4-Stream interfaces present	RO	Configuration dependent
[25:24]	axdata_width	Data width on AXI/ACE-Lite interface: 2'b00: 128 bits 2'b01: 256 bits 2'b10: 512 bits	RO	Configuration dependent
[23:21]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[20:16]	num_wr_data_buf	Number of write data buffers in HN-I	RO	Configuration dependent
[15:8]	num_ax_reqs	Maximum number of outstanding AXI/ACE-Lite requests	RO	Configuration dependent
[7:0]	num_excl	Number of exclusive monitors in HN-I	RO	Configuration dependent

5.3.3.5 por_hni_sam_addrregion0_cfg

Configures HN-I SAM 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

Figure 5-43: por_hni_sam_addrregion0_cfg

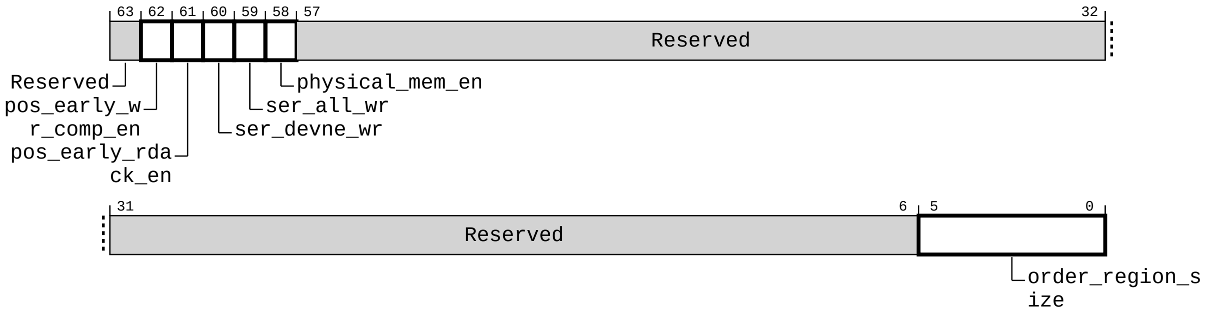


Table 5-59: por_hni_sam_addrregion0_cfg attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	pos_early_wr_comp_en	Enables early write acknowledgment in address region 0. This function is 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. This function is 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>; sets order region 0 size within address region 0 (2^n*4KB)	RW	6'b111111

5.3.3.6 por_hni_sam_addrregion1_cfg

Configures HN-I SAM 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

Figure 5-44: por_hni_sam_addrregion1_cfg

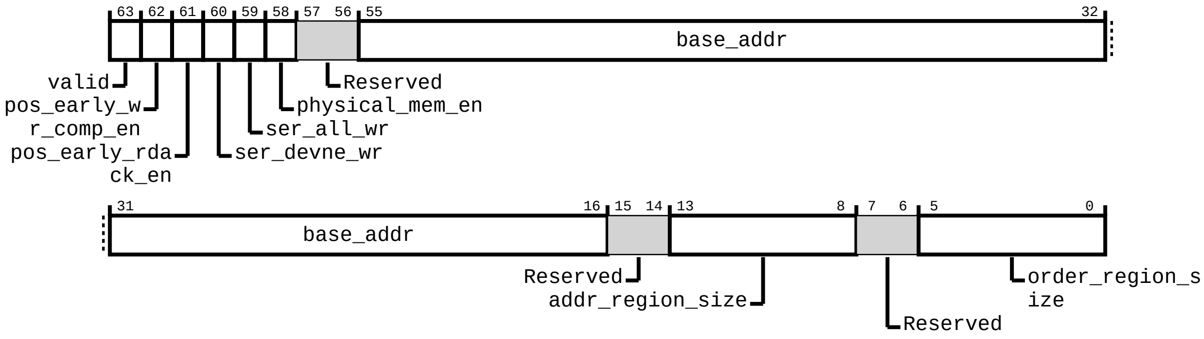


Table 5-60: por_hni_sam_addrregion1_cfg attributes

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. This function is 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. This function is 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 the 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	-
[13:8]	addr_region_size	<n>; sets 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>; sets order region 1 size within address region 1 (2^n*4KB)	RW	6'h0

5.3.3.7 por_hni_sam_addrregion2_cfg

Configures HN-I SAM 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

Figure 5-45: por_hni_sam_addrregion2_cfg

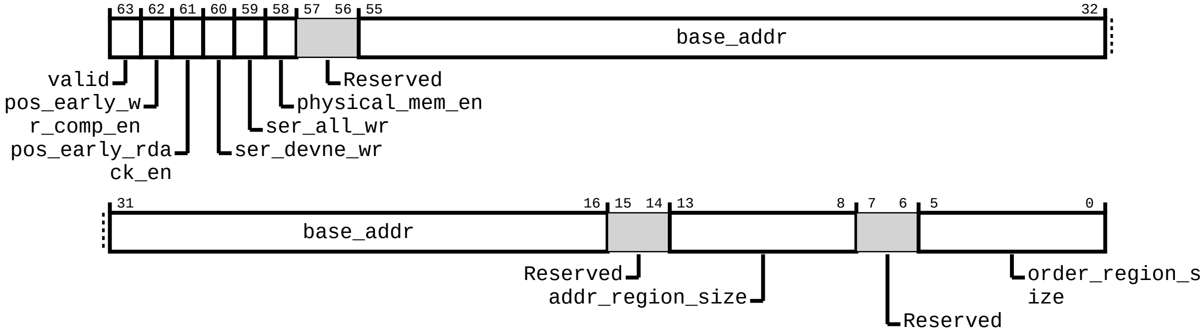


Table 5-61: por_hni_sam_addrregion2_cfg attributes

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. This function is 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. This function is 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 the Arm Architecture Reference Manual physical memory ordering guarantees.	RW	1'b0
[57:56]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[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>; sets 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>; sets order region 2 size within address region 2 (2^n*4KB)	RW	6'h0

5.3.3.8 por_hni_sam_addrregion3_cfg

Configures HN-I SAM 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

Figure 5-46: por_hni_sam_addrregion3_cfg

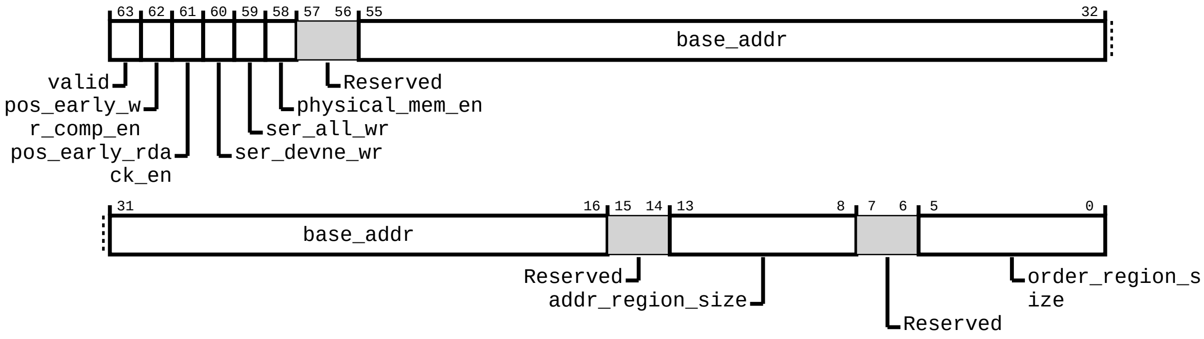


Table 5-62: por_hni_sam_addrregion3_cfg attributes

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. This function is 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. This function is 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 the 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	-
[13:8]	addr_region_size	<n>; sets 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>; sets order region 3 size within address region 3 (2^n*4KB)	RW	6'h0

5.3.3.9 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

Figure 5-47: por_hni_cfg_ctl

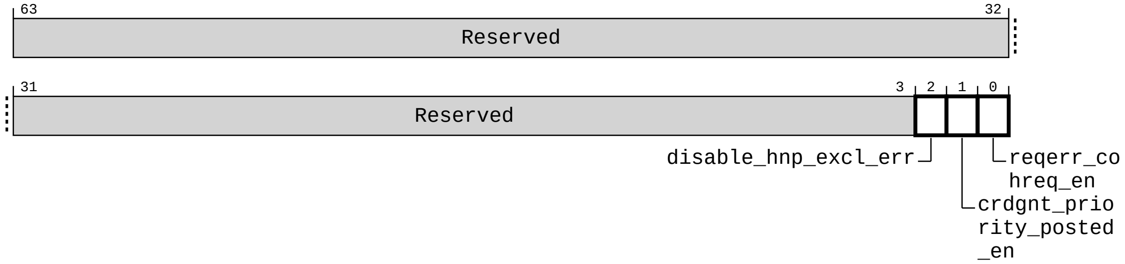


Table 5-63: por_hni_cfg_ctl attributes

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 or MakeUnique 3. Coherent or CopyBack write	RW	1'b1

5.3.3.10 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

Figure 5-48: por_hni_aux_ctl

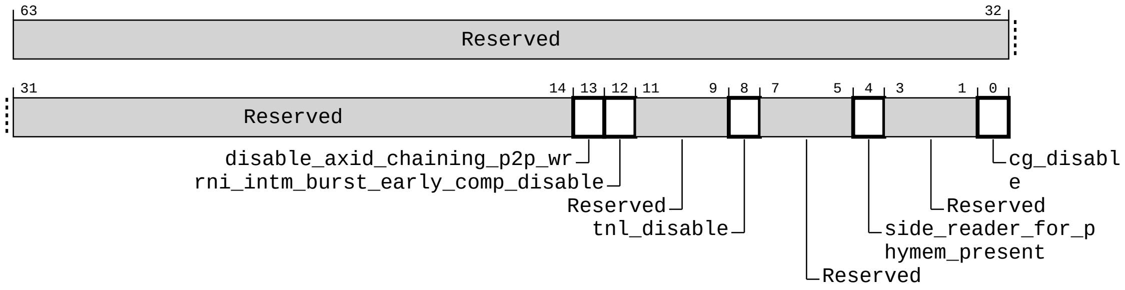

63
32
Reserved
31
14
13
12
11
9
8
7
5
4
3
1
0
Reserved
disable_axid_chaining_p2p_wr
rni_intm_burst_early_comp_disable
Reserved
tnl_disable
Reserved
side_reader_for_phymem_present
Reserved
cg_disable

Table 5-64: 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 the peer-to-peer write slice. This field only applies to HN-P nodes.	RW	1'b0
[12]	rni_intm_burst_early_comp_disable	Disables early COMP to RN-I for non-last burst writes	RW	1'b0
[11:9]	Reserved	Reserved	RO	-
[8]	tnl_disable	Disables RN-I-HN-I tunneling in HN-I. You must set por_rni_aux_ctl.dis_hni_wr_stream before setting this bit.	RW	1'b0
[7:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[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

5.3.3.11 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-49: por_hni_errfr

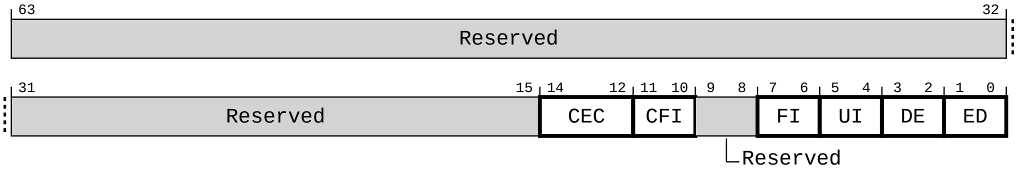


Table 5-65: por_hni_errfr attributes

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	-

Bits	Name	Description	Type	Reset
[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

5.3.3.12 por_hni_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-50: por_hni_errctlr

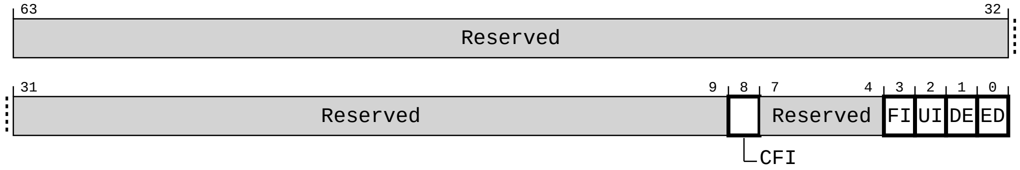


Table 5-66: por_hni_errctlr attributes

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

Bits	Name	Description	Type	Reset
[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

5.3.3.13 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-51: por_hni_errstatus

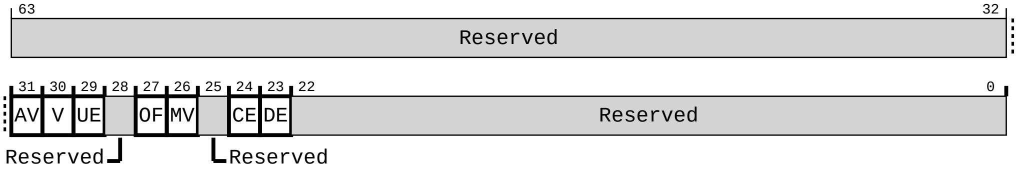


Table 5-67: por_hni_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[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 slave 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 the UE, DE, and 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	-

5.3.3.14 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-52: por_hni_erraddr

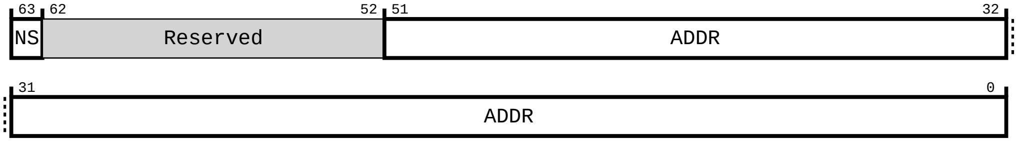


Table 5-68: por_hni_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of : 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

5.3.3.15 por_hni_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-53: por_hni_errmisc

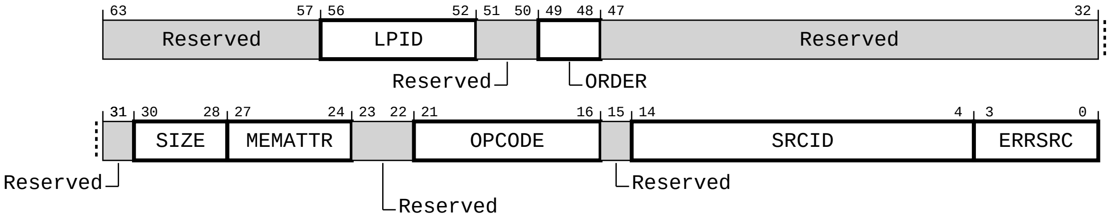


Table 5-69: por_hni_errmisc attributes

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
[23:22]	Reserved	Reserved	RO	-
[21:16]	OPCODE	Error opcode	RW	6'b0

Bits	Name	Description	Type	Reset
[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 or 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. This encoding only applies to HN-P nodes. Note: For configuration write data, BRESP, and poison errors, the only valid field is por_hni_errmisc.SRCID. For other error types, all fields are valid.	RW	4'b0

5.3.3.16 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

Figure 5-54: por_hni_errfr_NS

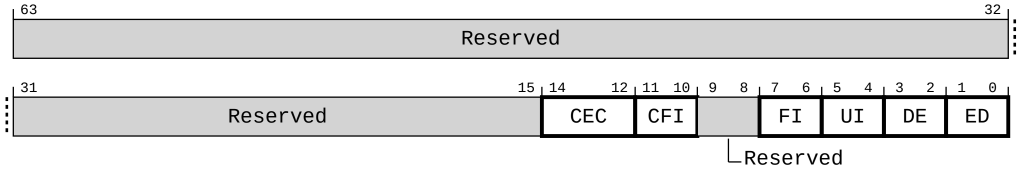


Table 5-70: por_hni_errfr_NS attributes

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

5.3.3.17 por_hni_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection and 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

Figure 5-55: por_hni_errctlr_NS

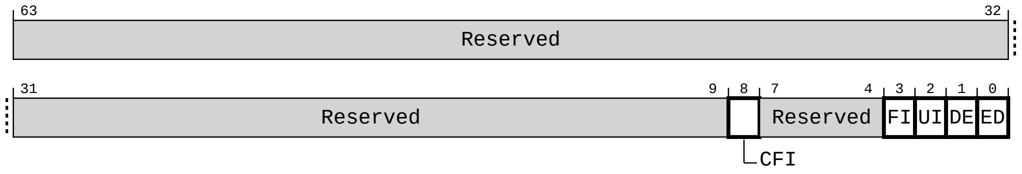


Table 5-71: por_hni_errctlr_NS attributes

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

5.3.3.18 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

Figure 5-56: por_hni_errstatus_NS

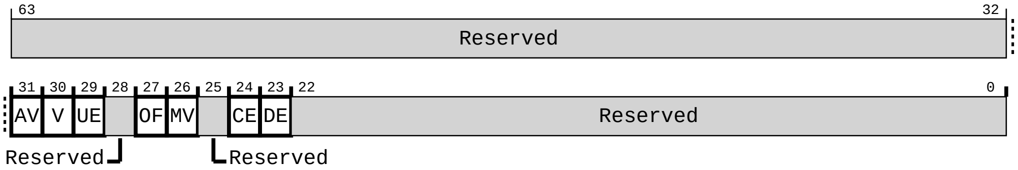


Table 5-72: por_hni_errstatus_NS attributes

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 is 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 slave 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 the UE, DE, and 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	-

5.3.3.19 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

Figure 5-57: por_hni_erraddr_NS

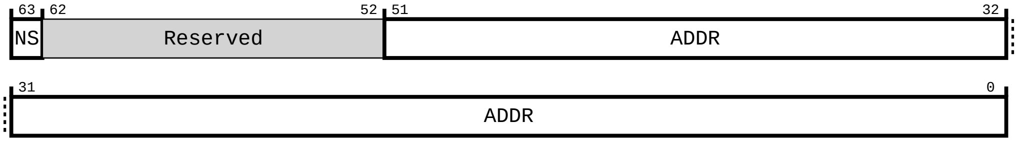


Table 5-73: por_hni_erraddr_NS attributes

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

5.3.3.20 por_hni_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred and 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

Figure 5-58: por_hni_errmisc_NS

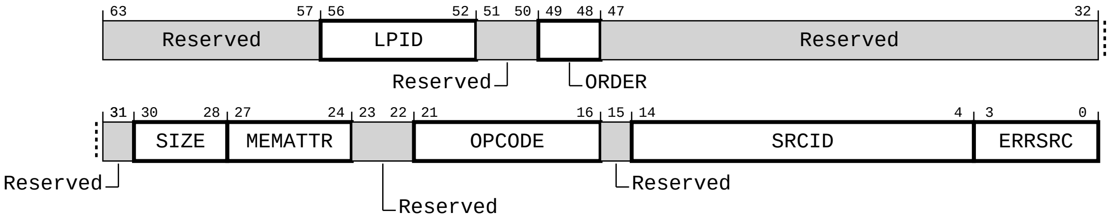


Table 5-74: por_hni_errmisc_NS attributes

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
[23:22]	Reserved	Reserved	RO	-
[21:16]	OPCODE	Error opcode	RW	6'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 or 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. This encoding only applies to HN-P nodes. Note: For configuration write data, BRESP, and poison errors, the only valid field is por_hni_errmisc_NS.SRCID. For other error types, all fields are valid.	RW	4'b0

5.3.3.21 por_hni_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-59: por_hni_pmu_event_sel

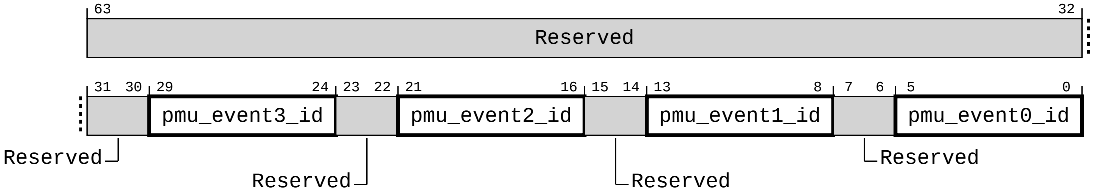


Table 5-75: por_hni_pmu_event_sel attributes

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, in other words 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

5.3.3.22 por_hnp_pmu_event_sel

Specifies the Performance Monitoring Unit (PMU) event to be counted. This register only applies to HN-P nodes.

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

Figure 5-60: por_hnp_pmu_event_sel

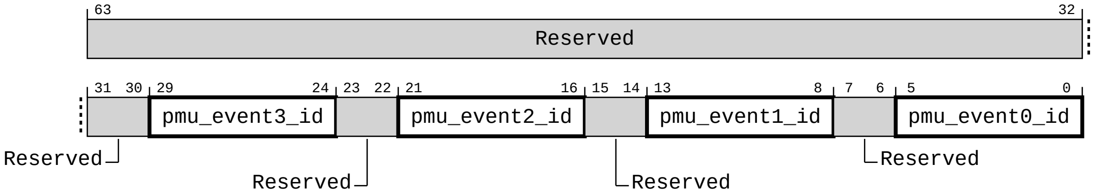


Table 5-76: por_hnp_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	Peer-to-peer slice PMU event 3 select. See pmu_event0_id for encodings.	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	Peer-to-peer slice PMU event 2 select. See pmu_event0_id for encodings.	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	Peer-to-peer slice 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	Peer-to-peer 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: AWVALID set 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

5.3.4 CXLA register descriptions

This section lists the CXLA registers.

5.3.4.1 por_cxla_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

Figure 5-61: por_cxla_node_info

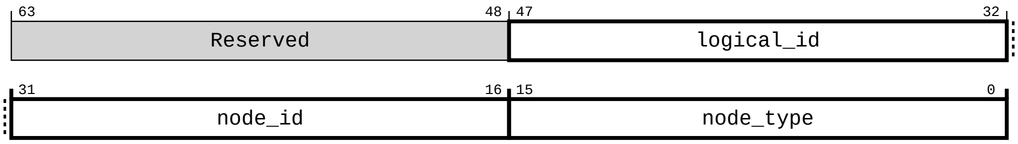


Table 5-77: por_cxla_node_info attributes

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-650 node type identifier	RO	16'h0102

5.3.4.2 por_cxla_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

Figure 5-62: por_cxla_child_info

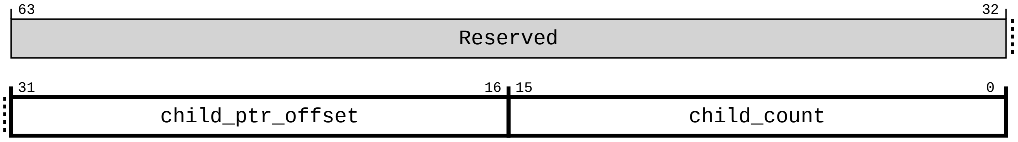


Table 5-78: por_cxla_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.4.3 por_cxla_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

Figure 5-63: por_cxla_secure_register_groups_override

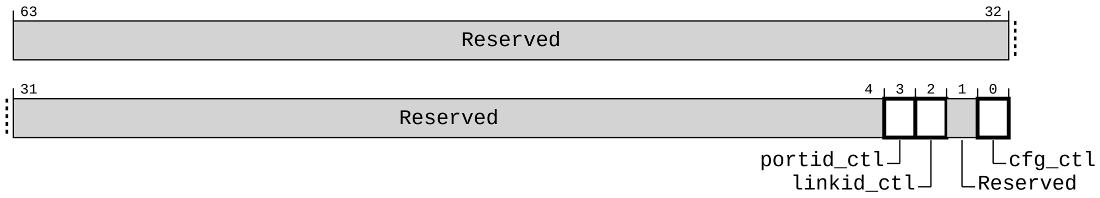


Table 5-79: por_cxla_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[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

5.3.4.4 por_cxla_unit_info

Provides component identification information for CXLA.

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

Figure 5-64: por_cxla_unit_info

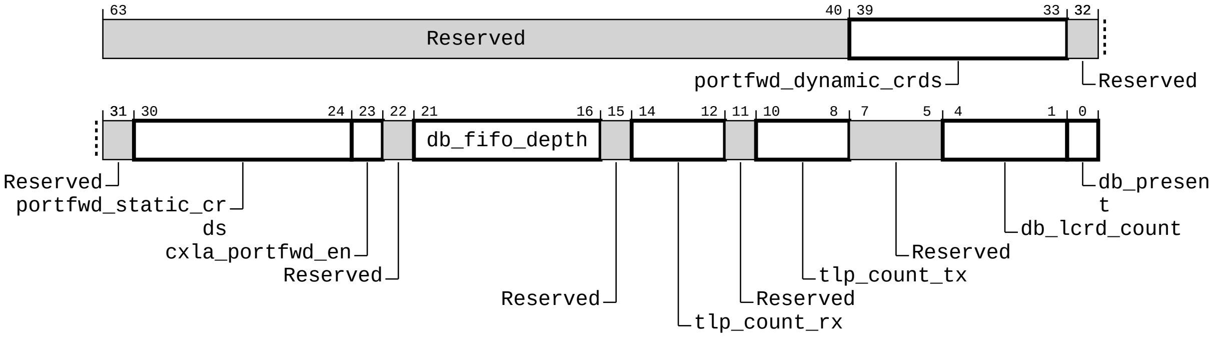


Table 5-80: por_cxla_unit_info attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:33]	portfwd_dynamic_crds	Number of dynamic credits that are granted by this CXLA port for port-forwarded traffic	RO	Configuration dependent
[32:31]	Reserved	Reserved	RO	-
[30:24]	portfwd_static_crds	Number of static credits that are granted by this CXLA port for port-forwarded traffic	RO	Configuration dependent
[23]	cxla_portfwd_en	Indicates whether port-to-port forwarding is enabled at this CXLA port	RO	Configuration dependent
[22]	Reserved	Reserved	RO	-
[21:16]	db_fifo_depth	FIFO depth in CXDB and PDB CXLA Domain Bridges	RO	Configuration dependent
[15]	Reserved	Reserved	RO	-
[14:12]	tlp_count_rx	Maximum number of TLPs supported by RX TLP buffer	RO	Configuration dependent
[11]	Reserved	Reserved	RO	-
[10:8]	tlp_count_tx	Maximum number of TLPs supported by TX TLP buffer	RO	Configuration dependent
[7:5]	Reserved	Reserved	RO	-
[4:1]	db_lcrd_count	Number of flit credits between CXG and CXLA	RO	Configuration dependent
[0]	db_present	DB present in CXLA	RO	Configuration dependent

5.3.4.5 por_cxla_cfg_ctl

Functions as the configuration control register for CXLA.

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_cxla_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

Figure 5-65: por_cxla_cfg_ctl

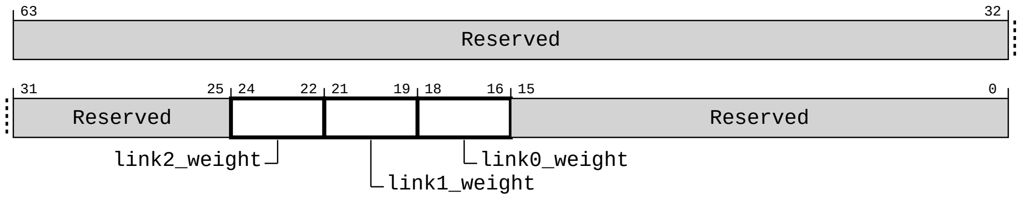


Table 5-81: por_cxla_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24:22]	link2_weight	Determines weight of link 2 to start forming a TLP in presence of pending messages to other links. Applies to message packing.	RW	3'b001
[21:19]	link1_weight	Determines weight of link 1 to start forming a TLP in presence of pending messages to other links. Applies to message packing.	RW	3'b001
[18:16]	link0_weight	Determines weight of link 0 to start forming a TLP in presence of pending messages to other links. Applies to message packing.	RW	3'b001
[15:0]	Reserved	Reserved	RO	-

5.3.4.6 por_cxla_aux_ctl

Functions as the auxiliary control register for CXLA.

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

Figure 5-66: por_cxla_aux_ctl

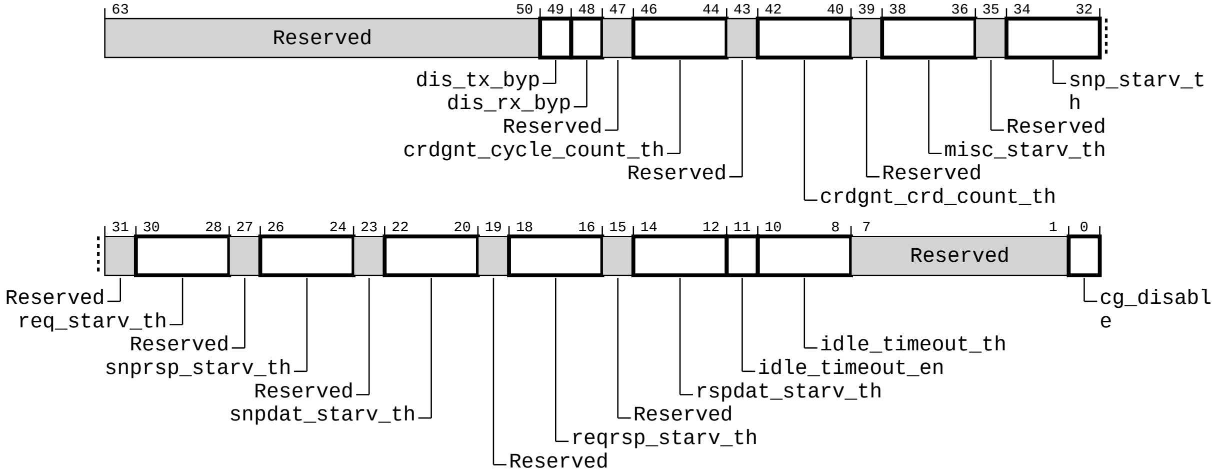


Table 5-82: por_cxla_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49]	dis_tx_byp	If set, disables TX bypass paths	RW	1'b0
[48]	dis_rx_byp	If set, disables RX bypass paths	RW	1'b0
[47]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[46:44]	crdgnt_cycle_count_th	Maximum number of cycles that must elapse from the end of previous TLP to send a 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 credits that must accumulate to send a credit grant message: 3'b000: 16 credits 3'b001: 32 credits 3'b010: 64 credits 3'b011: 128 credits	RW	3'b010
[39]	Reserved	Reserved	RO	-
[38:36]	misc_starv_th	Maximum number of consecutive instances a misc message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[35]	Reserved	Reserved	RO	-
[34:32]	snp_starv_th	Maximum number of consecutive instances a snoop request message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[31]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[30:28]	req_starv_th	Maximum number of consecutive instances a memory request message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[27]	Reserved	Reserved	RO	-
[26:24]	snprsp_starv_th	Maximum number of consecutive instances a snoop response without data message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[23]	Reserved	Reserved	RO	-
[22:20]	snpdat_starv_th	Maximum number of consecutive instances a snoop response with data message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[19]	Reserved	Reserved	RO	-
[18:16]	reqrsp_starv_th	Maximum number of consecutive instances a memory response without data message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:12]	rspdat_starv_th	Maximum number of consecutive instances a memory response with data message loses to other message types in forming a TLP: 3'b000: 8 cycles 3'b001: 16 cycles 3'b010: 32 cycles 3'b011: 64 cycles 3'b100: 128 cycles	RW	3'b010
[11]	idle_timeout_en	Enables idle timeout. Applies to message packing. If this bit is set, TLP packing continues until TLP length reaches the maximum configured or if the idle_timeout_th is reached.	RW	1'b0
[10:8]	idle_timeout_th	Maximum number of idle cycles a TLP waits for a message to be packed before ending the TLP. Applies to message packing: 3'b000: 4 cycles 3'b001: 8 cycles 3'b010: 16 cycles 3'b011: 32 cycles	RW	3'b001
[7:1]	Reserved	Reserved	RO	-
[0]	cg_disable	Disables CXLA architectural clock gates	RW	1'b0

5.3.4.7 por_cxla_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

Figure 5-67: por_cxla_ccix_prop_capabilities

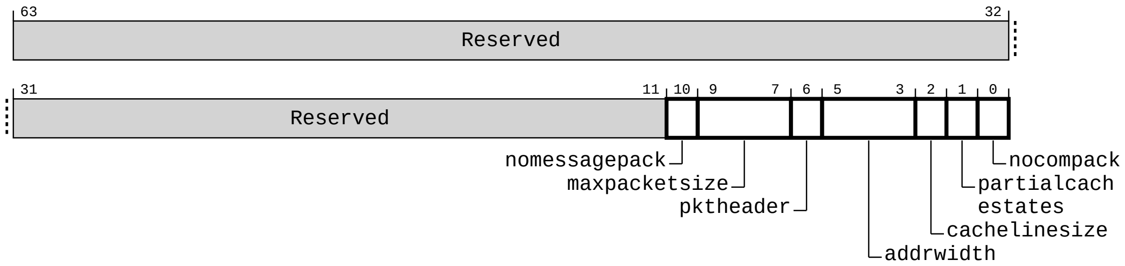


Table 5-83: por_cxla_ccix_prop_capabilities attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	nomessagepack	No message packing only supported: 1'b0: False 1'b1: True	RO	Configuration dependent
[9:7]	maxpacketsize	Maximum packet size supported: 3'b000: 128B 3'b001: 256B 3'b010: 512B	RO	Configuration dependent
[6]	pktheader	Packet header type supported: 1'b0: PCIe compatible header 1'b1: Optimized header	RO	Configuration dependent
[5:3]	addrwidth	Address width supported: 3'b000: 48b 3'b001: 52b 3'b010: 56b 3'b011: 60b 3'b100: 64b	RO	Configuration dependent
[2]	cachelinesize	Cacheline size supported: 1'b0: 64B 1'b1: 128B	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[1]	partialcachestates	Partial cache states supported: 1'b0: False 1'b1: True	RO	Configuration dependent
[0]	nocompack	No CompAck supported: 1'b0: False 1'b1: True	RO	Configuration dependent

5.3.4.8 por_cxla_ccix_prop_configured

Contains CCIX-configured properties.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC08

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

Figure 5-68: por_cxla_ccix_prop_configured

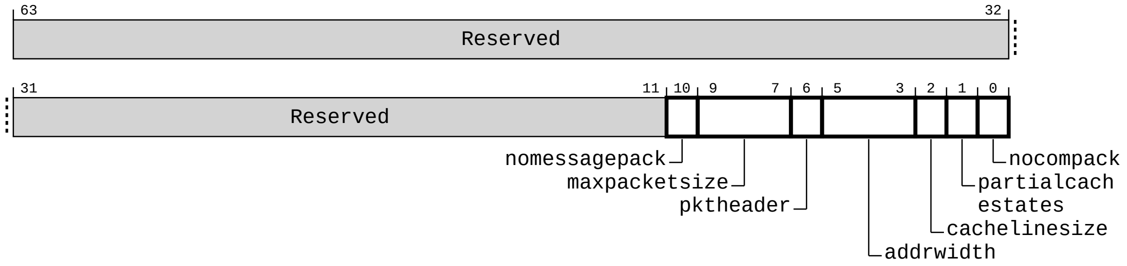


Table 5-84: por_cxla_ccix_prop_configured attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	nomessagepack	No message packing configured: 1'b0: False 1'b1: True	RW	1'b1
[9:7]	maxpacketsize	Maximum packet size configured: 3'b000: 128B 3'b001: 256B 3'b010: 512B	RW	3'b000
[6]	pktheader	Packet header configured: 1'b0: PCIe compatible header 1'b1: Optimized header	RW	1'b0
[5:3]	addrwidth	Address width configured: 3'b000: 48b 3'b001: 52b 3'b010: 56b 3'b011: 60b 3'b100: 64b	RW	3'b001
[2]	cachelinesize	Cacheline size configured: 1'b0: 64B 1'b1: 128B	RW	1'b0
[1]	partialcachestates	Partial cache states configured: 1'b0: False 1'b1: True	RW	1'b0
[0]	nocompack	No CompAck configured: 1'b0: False 1'b1: True	RW	1'b0

5.3.4.9 por_cxla_tx_cxs_attr_capabilities

Contains TX CXS supported attributes.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC10

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-69: por_cxla_tx_cxs_attr_capabilities

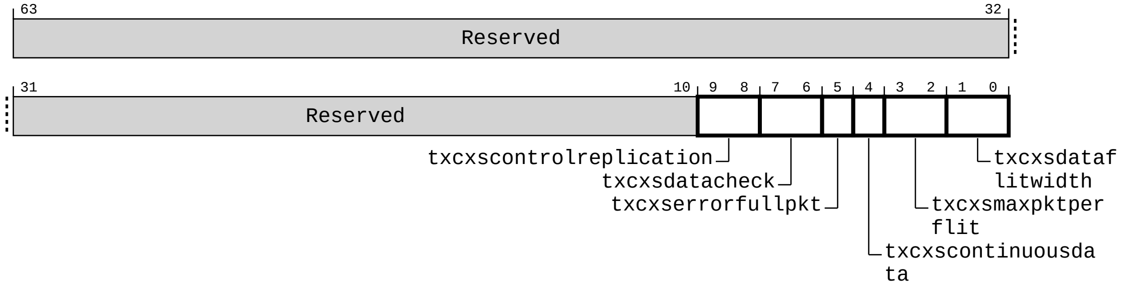


Table 5-85: por_cxla_tx_cxs_attr_capabilities attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:8]	txcxscontrolreplication	TX CXS control replication supported: 2'b00: None 2'b01: Duplicate 2'b10: Triplicate	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[7:6]	txcxsdatacheck	TX CXS Data Check supported: 2'b00: None 2'b01: Parity 2'b10: SECDED	RO	Configuration dependent
[5]	txcxserrorfullpkt	TX CXS error full packet supported: 1'b0: False 1'b1: True	RO	Configuration dependent
[4]	txcxscontinuousdata	TX CXS continuous data supported: 1'b0: False 1'b1: True	RO	Configuration dependent
[3:2]	txcxsmaxpktperflit	TX CXS maximum packets per flit supported: 2'b00: 2 2'b01: 3 2'b10: 4	RO	Configuration dependent
[1:0]	txcxsdataflitwidth	TX CXS data flit width supported: 2'b00: 256b 2'b01: 512b 2'b10: 1024b	RO	Configuration dependent

5.3.4.10 por_cxla_rx_cxs_attr_capabilities

Contains RX CXS supported attributes.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC18

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-70: por_cxla_rx_cxs_attr_capabilities

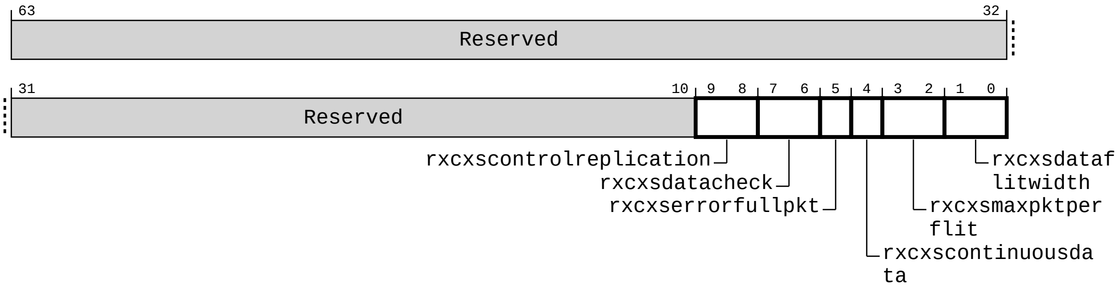


Table 5-86: por_cxla_rx_cxs_attr_capabilities attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:8]	rxcxscontrolreplication	RX CXS control replication supported: 2'b00: None 2'b01: Duplicate 2'b10: Triplicate	RO	Configuration dependent
[7:6]	rxcxsdatacheck	RX CXS Data Check supported: 2'b00: None 2'b01: Parity 2'b10: SECDED	RO	Configuration dependent
[5]	rxcxserrorfullpkt	RX CXS error full packet supported: 1'b0: False 1'b1: True	RO	Configuration dependent
[4]	rxcxscontinuousdata	RX CXS continuous data supported: 1'b0: False 1'b1: True	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[3:2]	rxcxsmaxpktperflit	RX CXS maximum packets per flit supported: 2'b00: 2 2'b01: 3 2'b10: 4	RO	Configuration dependent
[1:0]	rxcxsdataflitwidth	RX CXS data flit width supported: 2'b00: 256b 2'b01: 512b 2'b10: 1024b	RO	Configuration dependent

5.3.4.11 por_cxla_agentid_to_linkid_reg0

Specifies the mapping of Agent ID to Link ID for Agent IDs 0-7.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-71: por_cxla_agentid_to_linkid_reg0

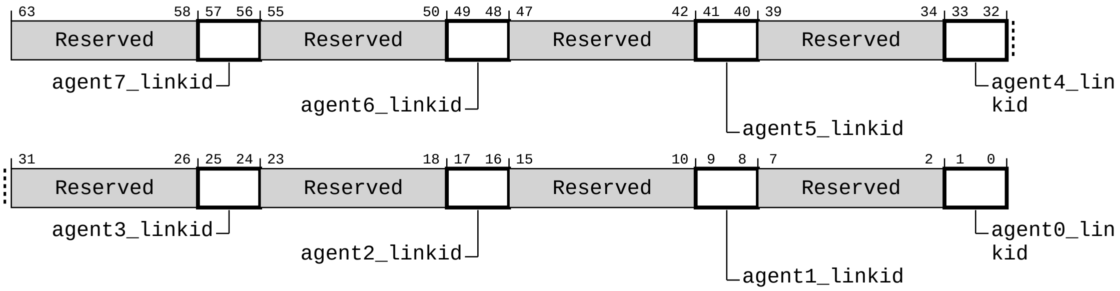


Table 5-87: por_cxla_agentid_to_linkid_reg0 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent7_linkid	Specifies the Link ID for Agent ID 7	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent6_linkid	Specifies the Link ID for Agent ID 6	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent5_linkid	Specifies the Link ID for Agent ID 5	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent4_linkid	Specifies the Link ID for Agent ID 4	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent3_linkid	Specifies the Link ID for Agent ID 3	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent2_linkid	Specifies the Link ID for Agent ID 2	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent1_linkid	Specifies the Link ID for Agent ID 1	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent0_linkid	Specifies the Link ID for Agent ID 0	RW	2'h0

5.3.4.12 por_cxla_agentid_to_linkid_reg1

Specifies the mapping of Agent ID to Link ID for Agent IDs 8-15.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-72: por_cxla_agentid_to_linkid_reg1

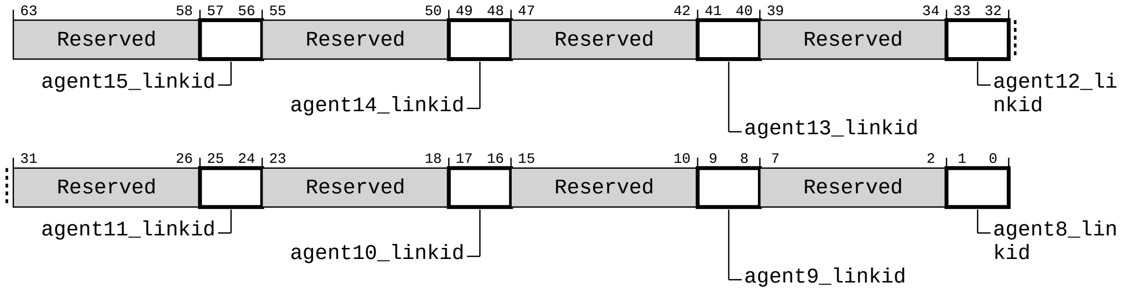


Table 5-88: por_cxla_agentid_to_linkid_reg1 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent15_linkid	Specifies the Link ID for Agent ID 15	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent14_linkid	Specifies the Link ID for Agent ID 14	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent13_linkid	Specifies the Link ID for Agent ID 13	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent12_linkid	Specifies the Link ID for Agent ID 12	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent11_linkid	Specifies the Link ID for Agent ID 11	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent10_linkid	Specifies the Link ID for Agent ID 10	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent9_linkid	Specifies the Link ID for Agent ID 9	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent8_linkid	Specifies the Link ID for Agent ID 8	RW	2'h0

5.3.4.13 por_cxla_agentid_to_linkid_reg2

Specifies the mapping of Agent ID to Link ID for Agent IDs 16-23.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-73: por_cxla_agentid_to_linkid_reg2

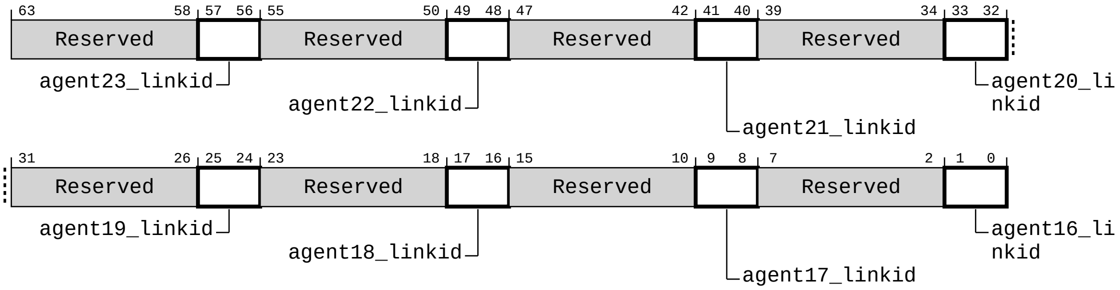


Table 5-89: por_cxla_agentid_to_linkid_reg2 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent23_linkid	Specifies the Link ID for Agent ID 23	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent22_linkid	Specifies the Link ID for Agent ID 22	RW	2'h0
[47:42]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[41:40]	agent21_linkid	Specifies the Link ID for Agent ID 21	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent20_linkid	Specifies the Link ID for Agent ID 20	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent19_linkid	Specifies the Link ID for Agent ID 19	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent18_linkid	Specifies the Link ID for Agent ID 18	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent17_linkid	Specifies the Link ID for Agent ID 17	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent16_linkid	Specifies the Link ID for Agent ID 16	RW	2'h0

5.3.4.14 por_cxla_agentid_to_linkid_reg3

Specifies the mapping of Agent ID to Link ID for Agent IDs 24-31.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-74: por_cxla_agentid_to_linkid_reg3

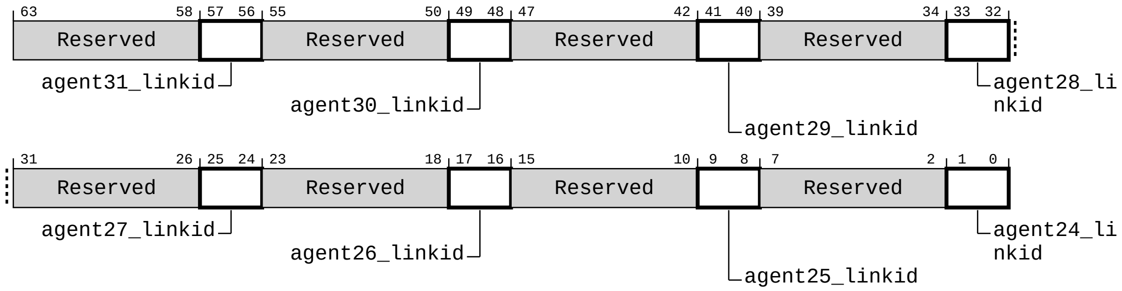


Table 5-90: por_cxla_agentid_to_linkid_reg3 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent31_linkid	Specifies the Link ID for Agent ID 31	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent30_linkid	Specifies the Link ID for Agent ID 30	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent29_linkid	Specifies the Link ID for Agent ID 29	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent28_linkid	Specifies the Link ID for Agent ID 28	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent27_linkid	Specifies the Link ID for Agent ID 27	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent26_linkid	Specifies the Link ID for Agent ID 26	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent25_linkid	Specifies the Link ID for Agent ID 25	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent24_linkid	Specifies the Link ID for Agent ID 24	RW	2'h0

5.3.4.15 por_cxla_agentid_to_linkid_reg4

Specifies the mapping of Agent ID to Link ID for Agent IDs 32-39.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-75: por_cxla_agentid_to_linkid_reg4

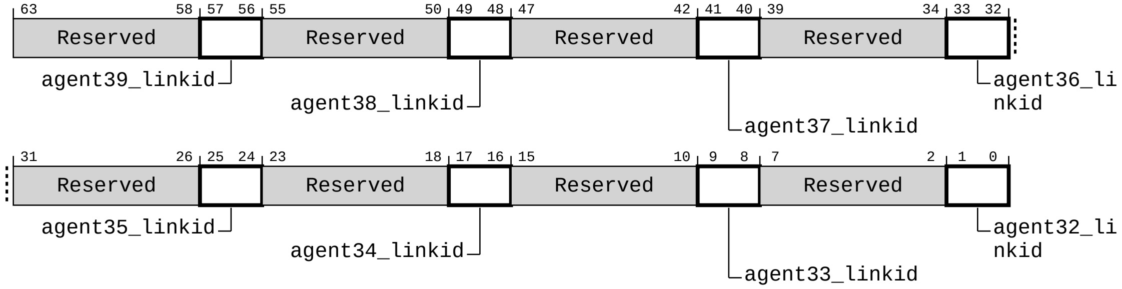


Table 5-91: por_cxla_agentid_to_linkid_reg4 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent39_linkid	Specifies the Link ID for Agent ID 39	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent38_linkid	Specifies the Link ID for Agent ID 38	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent37_linkid	Specifies the Link ID for Agent ID 37	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent36_linkid	Specifies the Link ID for Agent ID 36	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent35_linkid	Specifies the Link ID for Agent ID 35	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent34_linkid	Specifies the Link ID for Agent ID 34	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent33_linkid	Specifies the Link ID for Agent ID 33	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent32_linkid	Specifies the Link ID for Agent ID 32	RW	2'h0

5.3.4.16 por_cxla_agentid_to_linkid_reg5

Specifies the mapping of Agent ID to Link ID for Agent IDs 40-47.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-76: por_cxla_agentid_to_linkid_reg5

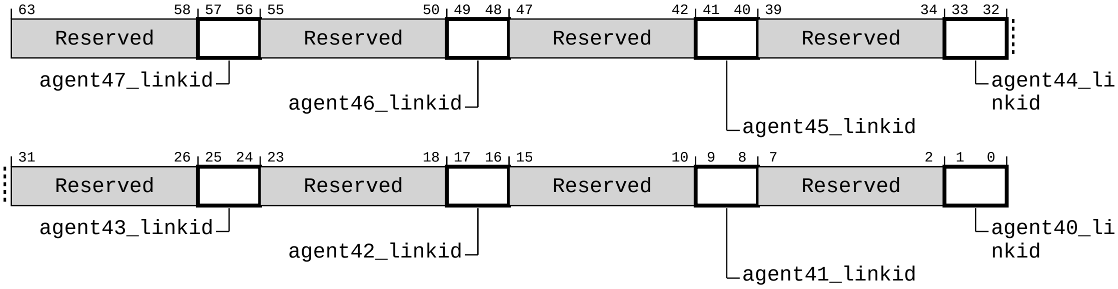


Table 5-92: por_cxla_agentid_to_linkid_reg5 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent47_linkid	Specifies the Link ID for Agent ID 47	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent46_linkid	Specifies the Link ID for Agent ID 46	RW	2'h0
[47:42]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[41:40]	agent45_linkid	Specifies the Link ID for Agent ID 45	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent44_linkid	Specifies the Link ID for Agent ID 44	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent43_linkid	Specifies the Link ID for Agent ID 43	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent42_linkid	Specifies the Link ID for Agent ID 42	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent41_linkid	Specifies the Link ID for Agent ID 41	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent40_linkid	Specifies the Link ID for Agent ID 40	RW	2'h0

5.3.4.17 por_cxla_agentid_to_linkid_reg6

Specifies the mapping of Agent ID to Link ID for Agent IDs 48-55.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-77: por_cxla_agentid_to_linkid_reg6

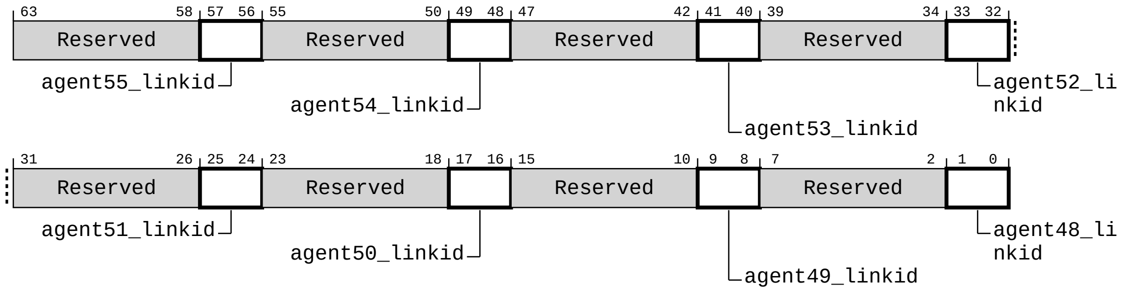


Table 5-93: por_cxla_agentid_to_linkid_reg6 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent55_linkid	Specifies the Link ID for Agent ID 55	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent54_linkid	Specifies the Link ID for Agent ID 54	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent53_linkid	Specifies the Link ID for Agent ID 53	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent52_linkid	Specifies the Link ID for Agent ID 52	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent51_linkid	Specifies the Link ID for Agent ID 51	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent50_linkid	Specifies the Link ID for Agent ID 50	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent49_linkid	Specifies the Link ID for Agent ID 49	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent48_linkid	Specifies the Link ID for Agent ID 48	RW	2'h0

5.3.4.18 por_cxla_agentid_to_linkid_reg7

Specifies the mapping of Agent ID to Link ID for Agent IDs 56-63.

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

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-78: por_cxla_agentid_to_linkid_reg7

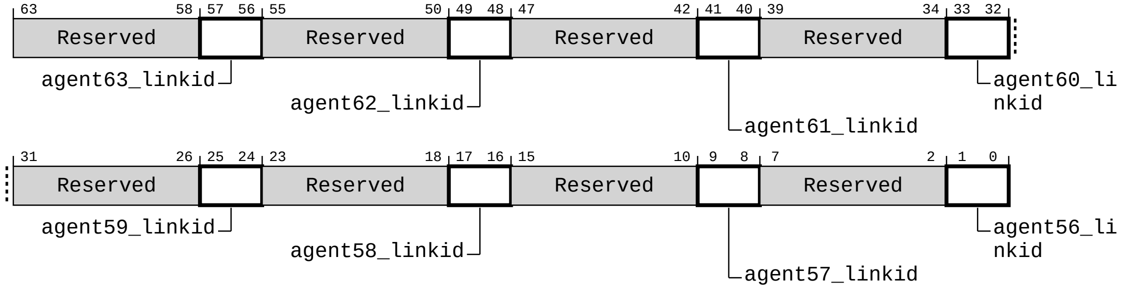


Table 5-94: por_cxla_agentid_to_linkid_reg7 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:56]	agent63_linkid	Specifies the Link ID for Agent ID 63	RW	2'h0
[55:50]	Reserved	Reserved	RO	-
[49:48]	agent62_linkid	Specifies the Link ID for Agent ID 62	RW	2'h0
[47:42]	Reserved	Reserved	RO	-
[41:40]	agent61_linkid	Specifies the Link ID for Agent ID 61	RW	2'h0
[39:34]	Reserved	Reserved	RO	-
[33:32]	agent60_linkid	Specifies the Link ID for Agent ID 60	RW	2'h0
[31:26]	Reserved	Reserved	RO	-
[25:24]	agent59_linkid	Specifies the Link ID for Agent ID 59	RW	2'h0
[23:18]	Reserved	Reserved	RO	-
[17:16]	agent58_linkid	Specifies the Link ID for Agent ID 58	RW	2'h0
[15:10]	Reserved	Reserved	RO	-
[9:8]	agent57_linkid	Specifies the Link ID for Agent ID 57	RW	2'h0
[7:2]	Reserved	Reserved	RO	-
[1:0]	agent56_linkid	Specifies the Link ID for Agent ID 56	RW	2'h0

5.3.4.19 por_cxla_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'hC70

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxla_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-79: por_cxla_agentid_to_linkid_val

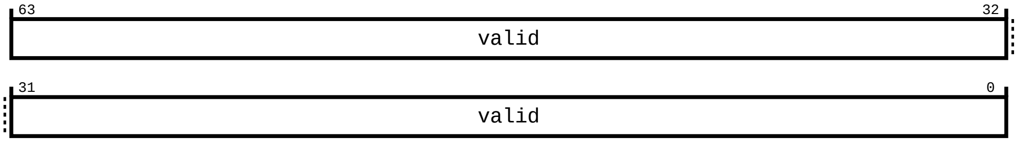


Table 5-95: por_cxla_agentid_to_linkid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Link ID is valid. The bit number corresponds to logical Agent ID number, from 0-63.	RW	63'h0

5.3.4.20 por_cxla_linkid_to_pcie_bus_num

Specifies the mapping of CCIX Link ID to PCIe bus number.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC78

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

Figure 5-80: por_cxla_linkid_to_pcie_bus_num

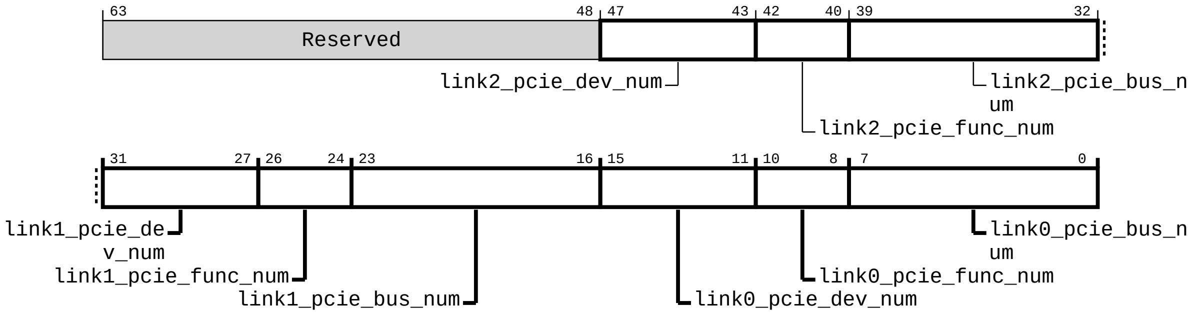


Table 5-96: por_cxla_linkid_to_pcie_bus_num attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:43]	link2_pcie_dev_num	PCIe Device number for Link ID 2	RW	5'h00
[42:40]	link2_pcie_func_num	PCIe Function number for Link ID 2	RW	3'h0
[39:32]	link2_pcie_bus_num	PCIe bus number for Link ID 2	RW	8'h00
[31:27]	link1_pcie_dev_num	PCIe Device number for Link ID 1	RW	5'h00
[26:24]	link1_pcie_func_num	PCIe Function number for Link ID 1	RW	3'h0
[23:16]	link1_pcie_bus_num	PCIe bus number for Link ID 1	RW	8'h00
[15:11]	link0_pcie_dev_num	PCIe Device number for Link ID 0	RW	5'h00
[10:8]	link0_pcie_func_num	PCIe Function number for Link ID 0	RW	3'h0
[7:0]	link0_pcie_bus_num	PCIe bus number for Link ID 0	RW	8'h00

5.3.4.21 por_cxla_tlp_hdr_fields

Configures TLP header field values.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC80

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

Figure 5-81: por_cxla_tlp_hdr_fields

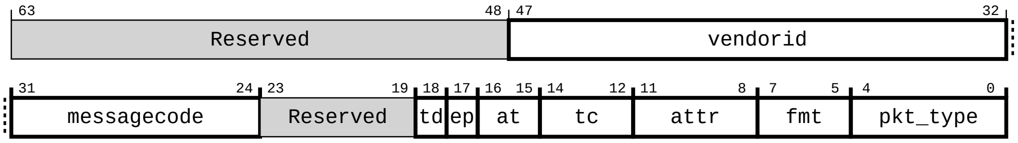


Table 5-97: por_cxla_tlp_hdr_fields attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	vendorid	Vendor ID	RW	16'b0
[31:24]	messagecode	Message code	RW	8'b01111111
[23:19]	Reserved	Reserved	RO	-
[18]	td	TLP digest	RW	1'b0
[17]	ep	Error forwarding	RW	1'b0
[16:15]	at	Address type	RW	2'b00
[14:12]	tc	Traffic class	RW	3'b000
[11:8]	attr	Attributes	RW	4'b0000
[7:5]	fmt	Format	RW	3'b011
[4:0]	pkt_type	Type	RW	5'b10010

5.3.4.22 por_cxla_permsg_pyld_0_63

Contains bits[63:0] of the 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

Figure 5-82: por_cxla_permsg_pyld_0_63

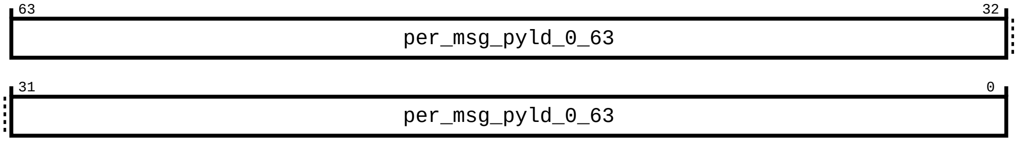


Table 5-98: por_cxla_permsg_pyld_0_63 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_0_63	PER message payload[63:0]	RW	64'b0

5.3.4.23 por_cxla_permsg_pyld_64_127

Contains bits[127:64] of the 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

Figure 5-83: por_cxla_permsg_pyld_64_127

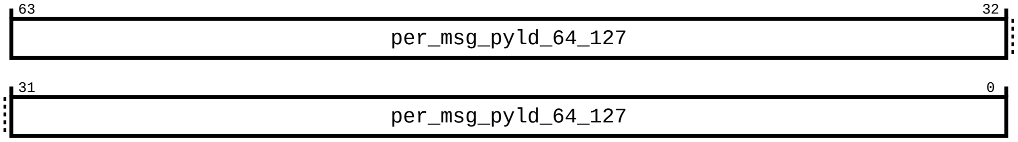


Table 5-99: por_cxla_permsg_pyld_64_127 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_64_127	PER message payload[127:64]	RW	64'b0

5.3.4.24 por_cxla_permsg_pyld_128_191

Contains bits[192:128] of the 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

Figure 5-84: por_cxla_permsg_pyld_128_191

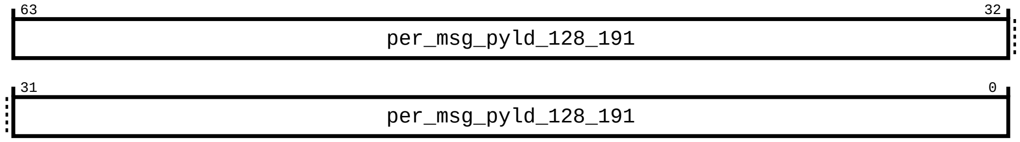


Table 5-100: por_cxla_permsg_pyld_128_191 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_128_191	PER message payload[191:128]	RW	64'b0

5.3.4.25 por_cxla_permsg_pyld_192_255

Contains bits[255:192] of the 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

Figure 5-85: por_cxla_permsg_pyld_192_255

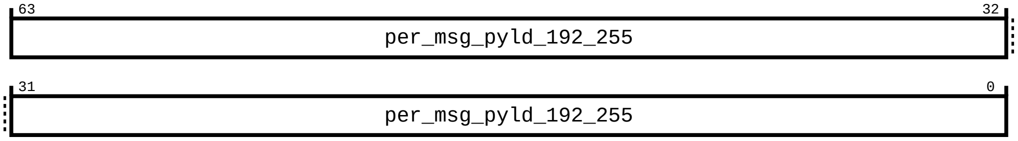


Table 5-101: por_cxla_permsg_pyld_192_255 attributes

Bits	Name	Description	Type	Reset
[63:0]	per_msg_pyld_192_255	PER message payload[255:192]	RW	64'b0

5.3.4.26 por_cxla_permsg_ctl

Contains control bits to trigger a 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

Figure 5-86: por_cxla_permsg_ctl

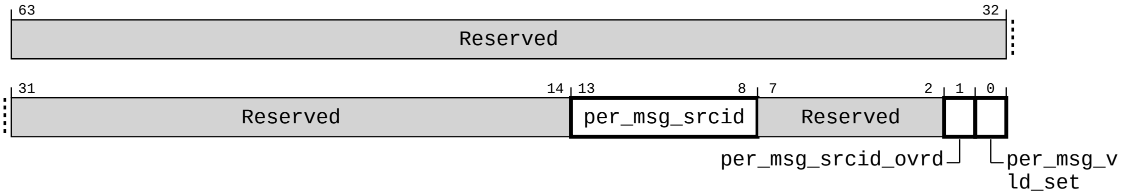


Table 5-102: por_cxla_permsg_ctl attributes

Bits	Name	Description	Type	Reset
[63:14]	Reserved	Reserved	RO	-
[13:8]	per_msg_srcid	Contains the Source ID that is used on CCIX PER message. Used when per_msg_srcid_ovrd is set.	RW	6'b0
[7:2]	Reserved	Reserved	RO	-
[1]	per_msg_srcid_ovrd	If set, overrides the Source ID on PER message with the value that is specified in this register. Otherwise, the source ID from payload[55:48] is used.	RW	1'b0
[0]	per_msg_vld_set	If set, sends CCIX PER message. This bit must be cleared after the current error is processed and before a new error message is triggered.	RW	1'b0

5.3.4.27 por_cxla_err_agent_id

Contains Error Agent ID. Must be programmed by the CCIX discovery software. Used as the TargetID on a 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

Figure 5-87: por_cxla_err_agent_id

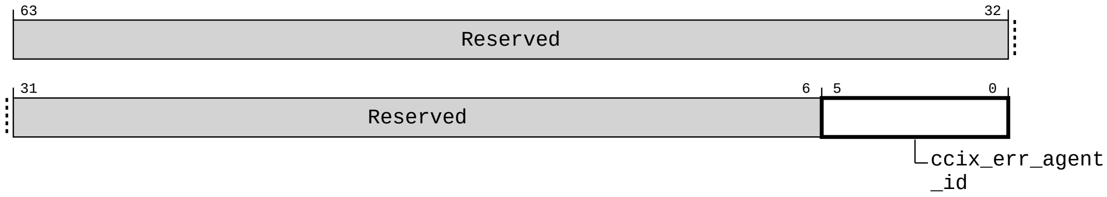


Table 5-103: por_cxla_err_agent_id attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:0]	ccix_err_agent_id	CCIX Error AgentID	RW	6'b0

5.3.4.28 por_cxla_agentid_to_portid_reg0

Specifies the mapping of Agent ID to Port ID for Agent IDs 0-7.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-88: por_cxla_agentid_to_portid_reg0

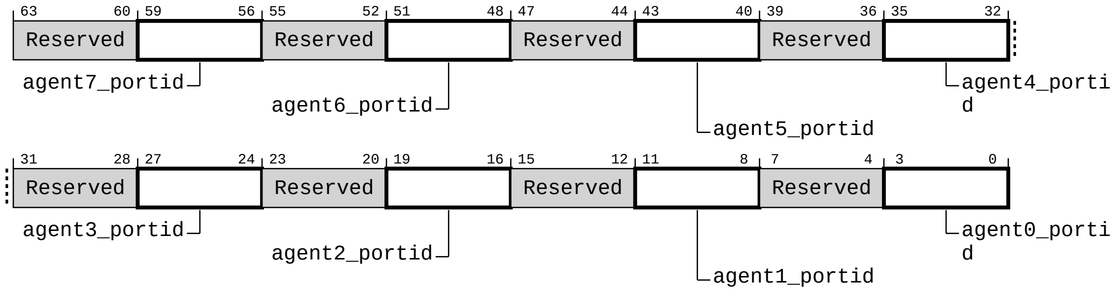


Table 5-104: por_cxla_agentid_to_portid_reg0 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[59:56]	agent7_portid	Specifies the Port ID for Agent ID 7	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent6_portid	Specifies the Port ID for Agent ID 6	RW	4'h0
[47:44]	Reserved	Reserved	RO	-
[43:40]	agent5_portid	Specifies the Port ID for Agent ID 5	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent4_portid	Specifies the Port ID for Agent ID 4	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent3_portid	Specifies the Port ID for Agent ID 3	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent2_portid	Specifies the Port ID for Agent ID 2	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent1_portid	Specifies the Port ID for Agent ID 1	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent0_portid	Specifies the Port ID for Agent ID 0	RW	4'h0

5.3.4.29 por_cxla_agentid_to_portid_reg1

Specifies the mapping of Agent ID to Port ID for Agent IDs 8-15.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-89: por_cxla_agentid_to_portid_reg1

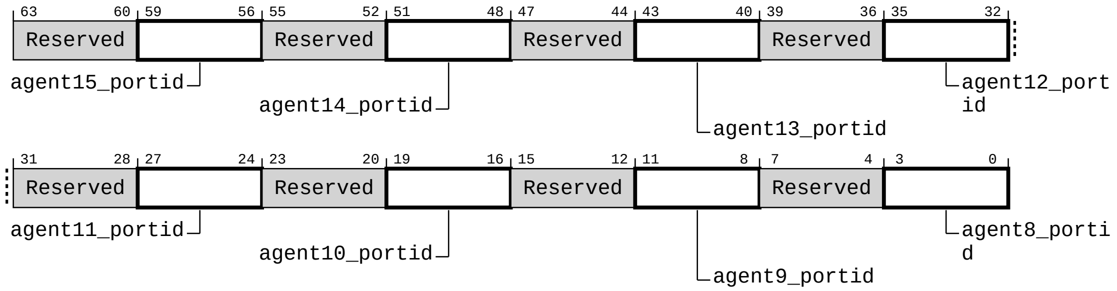


Table 5-105: por_cxla_agentid_to_portid_reg1 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent15_portid	Specifies the Port ID for Agent ID 15	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent14_portid	Specifies the Port ID for Agent ID 14	RW	4'h0
[47:44]	Reserved	Reserved	RO	-
[43:40]	agent13_portid	Specifies the Port ID for Agent ID 13	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent12_portid	Specifies the Port ID for Agent ID 12	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent11_portid	Specifies the Port ID for Agent ID 11	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent10_portid	Specifies the Port ID for Agent ID 10	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent9_portid	Specifies the Port ID for Agent ID 9	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent8_portid	Specifies the Port ID for Agent ID 8	RW	4'h0

5.3.4.30 por_cxla_agentid_to_portid_reg2

Specifies the mapping of Agent ID to Port ID for Agent IDs 16-23.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-90: por_cxla_agentid_to_portid_reg2

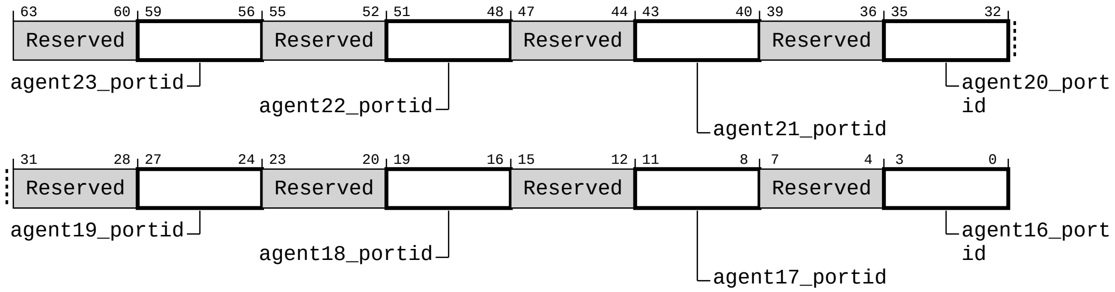


Table 5-106: por_cxla_agentid_to_portid_reg2 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent23_portid	Specifies the Port ID for Agent ID 23	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent22_portid	Specifies the Port ID for Agent ID 22	RW	4'h0
[47:44]	Reserved	Reserved	RO	-
[43:40]	agent21_portid	Specifies the Port ID for Agent ID 21	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent20_portid	Specifies the Port ID for Agent ID 20	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent19_portid	Specifies the Port ID for Agent ID 19	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent18_portid	Specifies the Port ID for Agent ID 18	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent17_portid	Specifies the Port ID for Agent ID 17	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent16_portid	Specifies the Port ID for Agent ID 16	RW	4'h0

5.3.4.31 por_cxla_agentid_to_portid_reg3

Specifies the mapping of Agent ID to Port ID for Agent IDs 24-31.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-91: por_cxla_agentid_to_portid_reg3

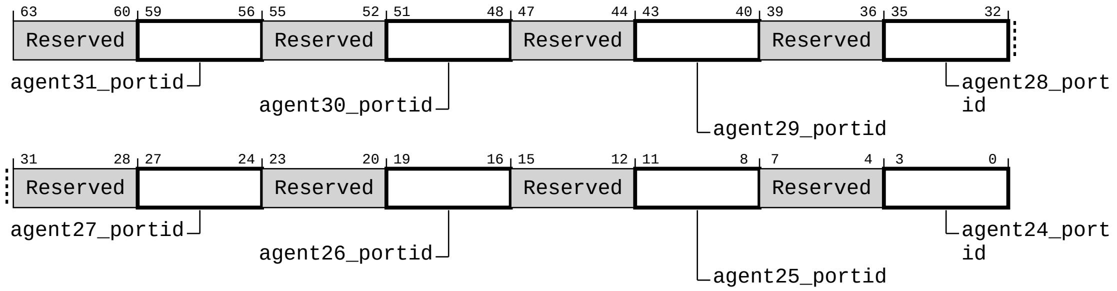


Table 5-107: por_cxla_agentid_to_portid_reg3 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent31_portid	Specifies the Port ID for Agent ID 31	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent30_portid	Specifies the Port ID for Agent ID 30	RW	4'h0
[47:44]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[43:40]	agent29_portid	Specifies the Port ID for Agent ID 29	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent28_portid	Specifies the Port ID for Agent ID 28	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent27_portid	Specifies the Port ID for Agent ID 27	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent26_portid	Specifies the Port ID for Agent ID 26	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent25_portid	Specifies the Port ID for Agent ID 25	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent24_portid	Specifies the Port ID for Agent ID 24	RW	4'h0

5.3.4.32 por_cxla_agentid_to_portid_reg4

Specifies the mapping of Agent ID to Port ID for Agent IDs 32-39.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-92: por_cxla_agentid_to_portid_reg4

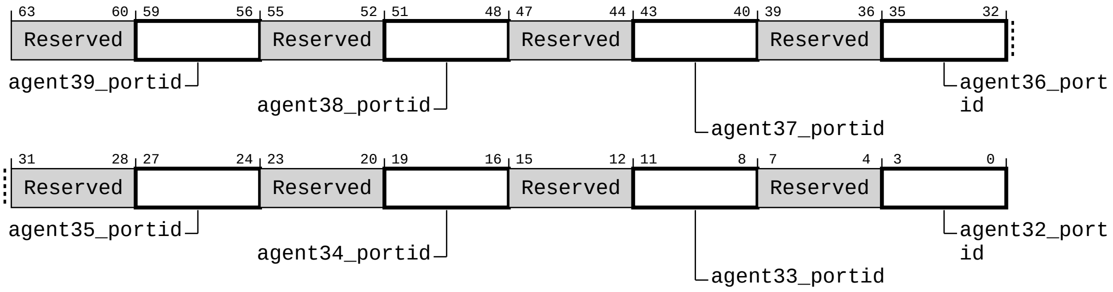


Table 5-108: por_cxla_agentid_to_portid_reg4 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent39_portid	Specifies the Port ID for Agent ID 39	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent38_portid	Specifies the Port ID for Agent ID 38	RW	4'h0
[47:44]	Reserved	Reserved	RO	-
[43:40]	agent37_portid	Specifies the Port ID for Agent ID 37	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent36_portid	Specifies the Port ID for Agent ID 36	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent35_portid	Specifies the Port ID for Agent ID 35	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent34_portid	Specifies the Port ID for Agent ID 34	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent33_portid	Specifies the Port ID for Agent ID 33	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent32_portid	Specifies the Port ID for Agent ID 32	RW	4'h0

5.3.4.33 por_cxla_agentid_to_portid_reg5

Specifies the mapping of Agent ID to Port ID for Agent IDs 40-47.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-93: por_cxla_agentid_to_portid_reg5

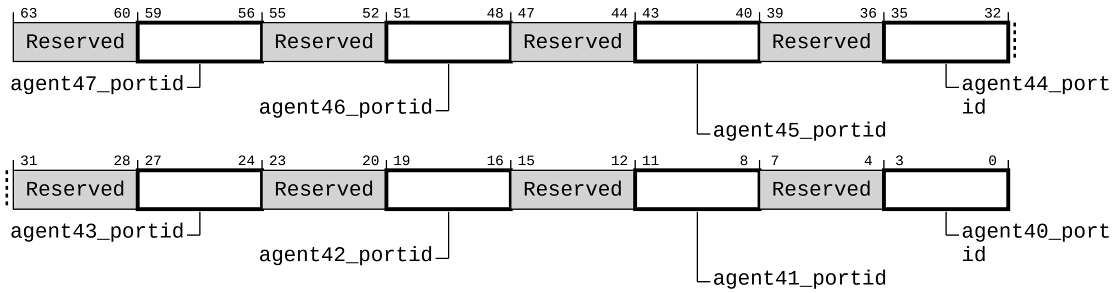


Table 5-109: por_cxla_agentid_to_portid_reg5 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent47_portid	Specifies the Port ID for Agent ID 47	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent46_portid	Specifies the Port ID for Agent ID 46	RW	4'h0
[47:44]	Reserved	Reserved	RO	-
[43:40]	agent45_portid	Specifies the Port ID for Agent ID 45	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent44_portid	Specifies the Port ID for Agent ID 44	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent43_portid	Specifies the Port ID for Agent ID 43	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent42_portid	Specifies the Port ID for Agent ID 42	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent41_portid	Specifies the Port ID for Agent ID 41	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent40_portid	Specifies the Port ID for Agent ID 40	RW	4'h0

5.3.4.34 por_cxla_agentid_to_portid_reg6

Specifies the mapping of Agent ID to Port ID for Agent IDs 48-55.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-94: por_cxla_agentid_to_portid_reg6

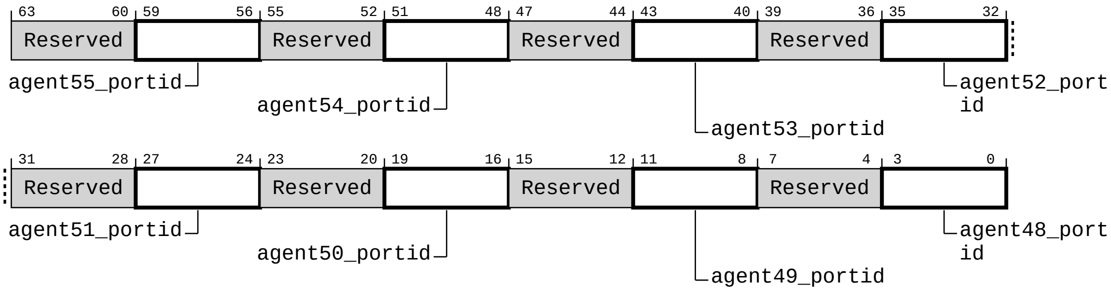


Table 5-110: por_cxla_agentid_to_portid_reg6 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent55_portid	Specifies the Port ID for Agent ID 55	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent54_portid	Specifies the Port ID for Agent ID 54	RW	4'h0
[47:44]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[43:40]	agent53_portid	Specifies the Port ID for Agent ID 53	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent52_portid	Specifies the Port ID for Agent ID 52	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent51_portid	Specifies the Port ID for Agent ID 51	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent50_portid	Specifies the Port ID for Agent ID 50	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent49_portid	Specifies the Port ID for Agent ID 49	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent48_portid	Specifies the Port ID for Agent ID 48	RW	4'h0

5.3.4.35 por_cxla_agentid_to_portid_reg7

Specifies the mapping of Agent ID to Port ID for Agent IDs 56-63.

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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-95: por_cxla_agentid_to_portid_reg7

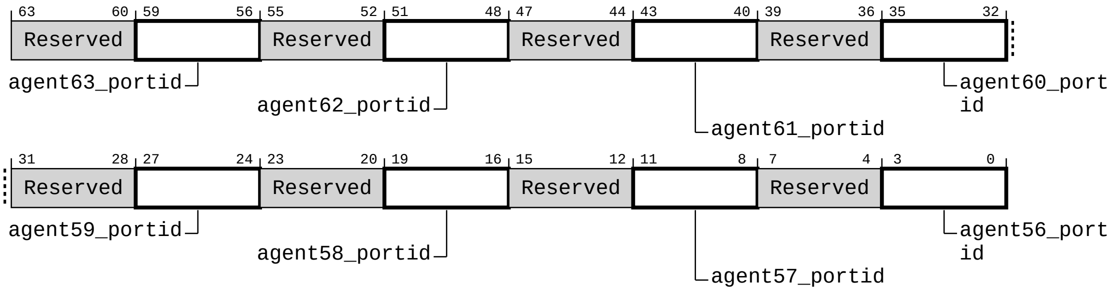


Table 5-111: por_cxla_agentid_to_portid_reg7 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:56]	agent63_portid	Specifies the Port ID for Agent ID 63	RW	4'h0
[55:52]	Reserved	Reserved	RO	-
[51:48]	agent62_portid	Specifies the Port ID for Agent ID 62	RW	4'h0
[47:44]	Reserved	Reserved	RO	-
[43:40]	agent61_portid	Specifies the Port ID for Agent ID 61	RW	4'h0
[39:36]	Reserved	Reserved	RO	-
[35:32]	agent60_portid	Specifies the Port ID for Agent ID 60	RW	4'h0
[31:28]	Reserved	Reserved	RO	-
[27:24]	agent59_portid	Specifies the Port ID for Agent ID 59	RW	4'h0
[23:20]	Reserved	Reserved	RO	-
[19:16]	agent58_portid	Specifies the Port ID for Agent ID 58	RW	4'h0
[15:12]	Reserved	Reserved	RO	-
[11:8]	agent57_portid	Specifies the Port ID for Agent ID 57	RW	4'h0
[7:4]	Reserved	Reserved	RO	-
[3:0]	agent56_portid	Specifies the Port ID for Agent ID 56	RW	4'h0

5.3.4.36 por_cxla_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_cxla_secure_register_groups_override.portid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-96: por_cxla_agentid_to_portid_val

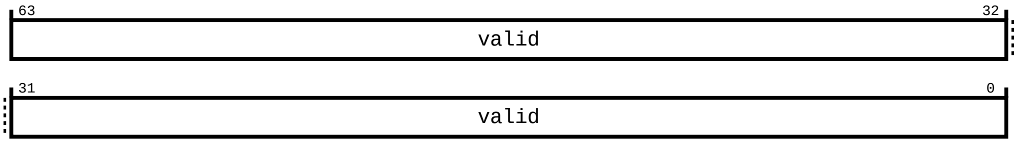


Table 5-112: por_cxla_agentid_to_portid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Port ID is valid. The bit number corresponds to logical Agent ID number, from 0-63.	RW	63'h0

5.3.4.37 por_cxla_portfwd_ctl

Functions as the port-to-port forwarding control register. Works with the por_cxla_portfwd_status register.

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

Figure 5-97: por_cxla_portfwd_ctl

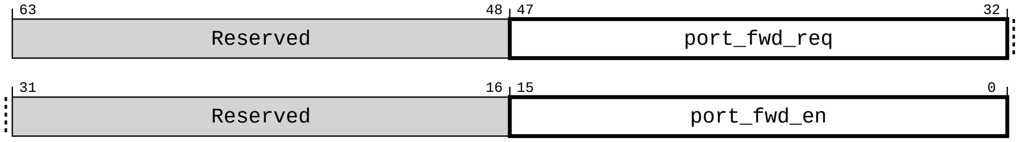


Table 5-113: por_cxla_portfwd_ctl attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	port_fwd_req	Bit vector, where each bit represents the logical Port ID of a port, in other words a CXG, that is present on CMN. Each bit is used to control the communication channel with the corresponding port: 1'b0: Port-to-port forwarding channel deactivate request 1'b1: Port-to-port forwarding channel activate request	RW	16'b0
[31:16]	Reserved	Reserved	RO	-
[15:0]	port_fwd_en	Bit vector, where each bit represents the logical Port ID of a port, in other words a CXG, that is present on CMN. Each bit, if 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	16'b0

5.3.4.38 por_cxla_portfwd_status

Functions as the port-to-port forwarding status register. Works with the por_cxla_portfwd_ctl register.

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

Figure 5-98: por_cxla_portfwd_status

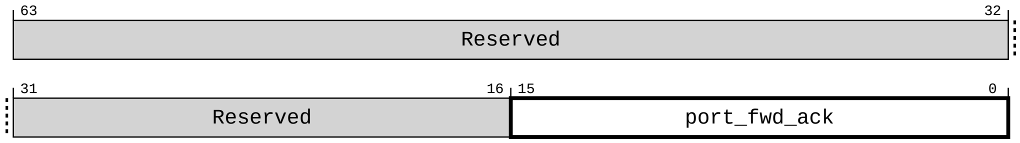


Table 5-114: por_cxla_portfwd_status attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	port_fwd_ack	Bit vector, where each bit represents the logical Port ID of a port, in other words a CXG, that is 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 inactive. 1'b1: Port-to-port forwarding channel is active	RO	16'b0

5.3.4.39 por_cxla_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-99: por_cxla_pmu_event_sel

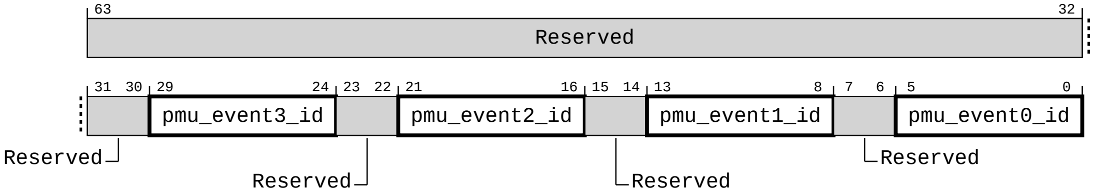


Table 5-115: por_cxla_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	CXLA PMU event 3 ID. See pmu_event0_id for encodings.	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	CXLA PMU event 2 ID. See pmu_event0_id for encodings.	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	CXLA 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	CXLA PMU event 0 ID: 6'h00: No event 6'h01: Average RX TLP count for Link 0 6'h02: Average RX TLP count for Link 1 6'h03: Average RX TLP count for Link 2 6'h04: Average TX TLP count for Link 0 6'h05: Average TX TLP count for Link 1 6'h06: Average TX TLP count for Link 2 6'h07: RX CXS for Link 0 6'h08: RX CXS for Link 1 6'h09: RX CXS for Link 2 6'h0A: TX CXS for Link 0 6'h0B: TX CXS for Link 1 6'h0C: TX CXS for Link 2 6'h0D: Average RX TLP size in DWs 6'h0E: Average TX TLP size in DWs 6'h0F: Average RX TLP size in CCIX messages 6'h10: Average TX TLP size in CCIX messages 6'h11: Average size of RX CXS in DWs within a beat 6'h12: Average size of TX CXS in DWs within a beat 6'h13: TX CXS link credit backpressure 6'h14: RX TLP buffer full and backpressured 6'h15: TX TLP buffer full and backpressured 6'h16: Reserved 6'h17: Average latency to form a TX TLP 6'h18: TX Request Chain 6'h19: RX RSPDAT (memory response with data) CGL buffer backpressured	RW	6'b0

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	6'h1A: RX SNPDAT (snoop response with data) CGL buffer backpressured 6'h1B: RX REQDAT (memory request) CGL buffer backpressured 6'h1C: RX REQRSP (memory response without data) CGL buffer backpressured 6'h1D: RX SNPRSP (snoop response without data) CGL buffer backpressured 6'h1E: RX SNP (snoop request) CGL buffer backpressured 6'h1F: RX UCMISC (misc) CGL buffer backpressured 6'h20: RX Number of port-forwarded CXS beats 6'h21: TX Number of port-forwarded CXS beats 6'h22: RX Number of port-forwarded message stalls due to static credits 6'h23: RX Number of port-forwarded message stalls due to dynamic credits	RW	6'b0

5.3.4.40 por_cxla_pmu_config

Configures the CXLA Performance Monitoring Unit (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

Figure 5-100: por_cxla_pmu_config

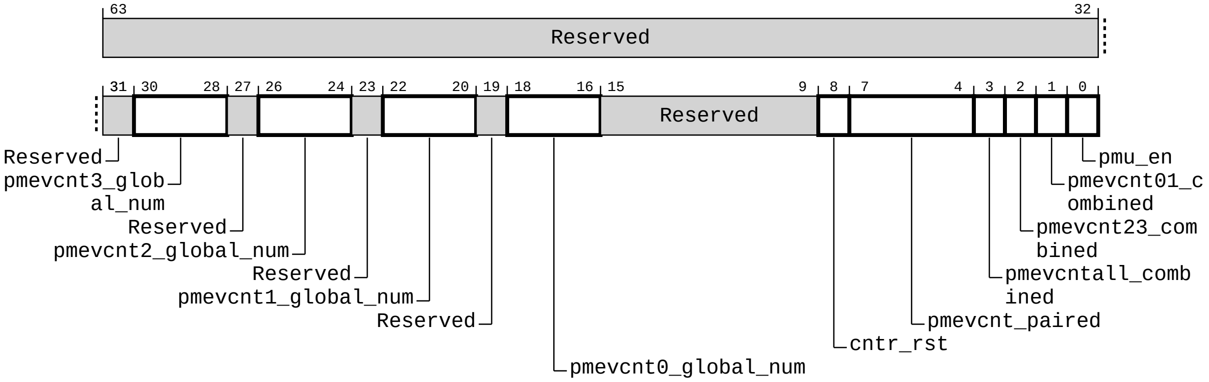


Table 5-116: por_cxla_pmu_config attributes

Bits	Name	Description	Type	Reset
[63:31]	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
[7:4]	pmevcnt_paired	PMU local counter is paired with global counter	RW	4'b0
[3]	pmevcntall_combined	Enables combination of all PMU counters, 0, 1, 2, and 3 NOTE: If 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

Bits	Name	Description	Type	Reset
[1]	pmevcnt01_combined	Enables combination of PMU counters 0 and 1	RW	1'b0
[0]	pmu_en	CXLA PMU enable Note: All other fields in this register are valid only if this bit is set.	RW	1'b0

5.3.4.41 por_cxla_pmevcnt

Contains all Performance Monitoring Unit (PMU) event counters, 0, 1, 2, and 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

Figure 5-101: por_cxla_pmevcnt

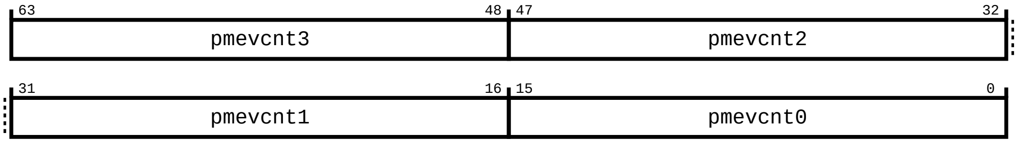


Table 5-117: por_cxla_pmevcnt attributes

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
[31:16]	pmevcnt1	PMU event counter 1	RW	16'h0000
[15:0]	pmevcnt0	PMU event counter 0	RW	16'h0000

5.3.4.42 por_cxla_pmevcntsr

Functions as the Performance Monitoring Unit (PMU) event counter shadow register for all PMU event counters, 0, 1, 2, and 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

Figure 5-102: por_cxla_pmevcntsr

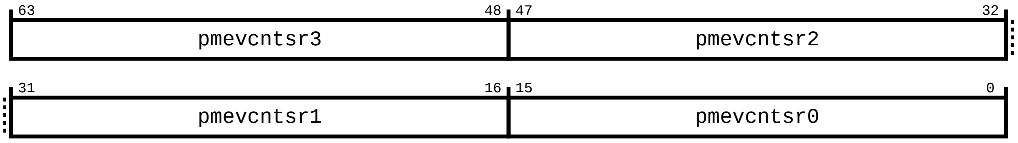


Table 5-118: por_cxla_pmevcntsr attributes

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

5.3.5 DN register descriptions

This section lists the DN registers.

5.3.5.1 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

Figure 5-103: por_dn_node_info

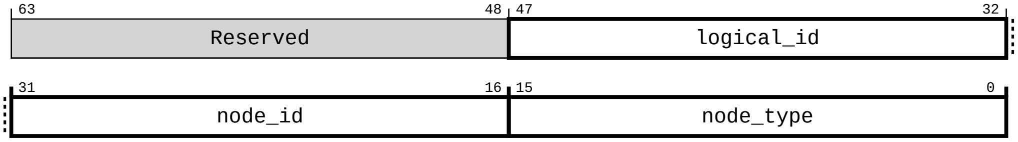


Table 5-119: por_dn_node_info attributes

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-650 node type identifier	RO	16'h0001

5.3.5.2 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

Figure 5-104: por_dn_child_info

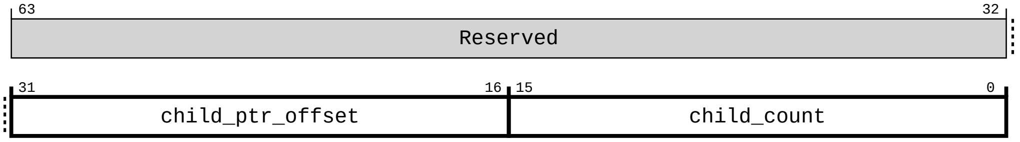


Table 5-120: por_dn_child_info attributes

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. This value is used in the discovery process.	RO	16'h0

5.3.5.3 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

Figure 5-105: por_dn_build_info

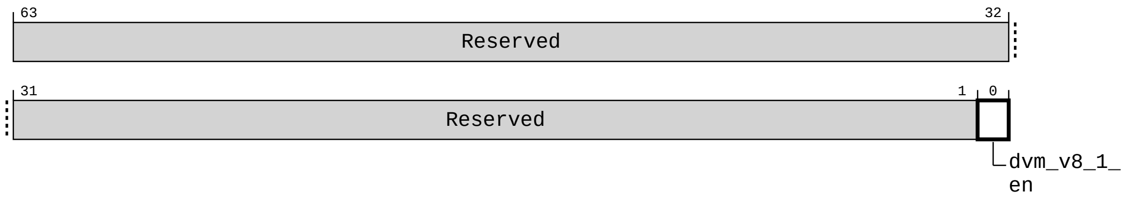


Table 5-121: por_dn_build_info attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	dvm_v8_1_en	Determines whether all nodes receiving DVM snoops support DVM v8.1 operations. You must set this bit to 0 if DVM v8.1 operations are not supported by all nodes. Setting this bit to 0 allows the node to perform demotion before sending out the DVM snoop.	RO	1'b1

5.3.5.4 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

Figure 5-106: por_dn_secure_register_groups_override

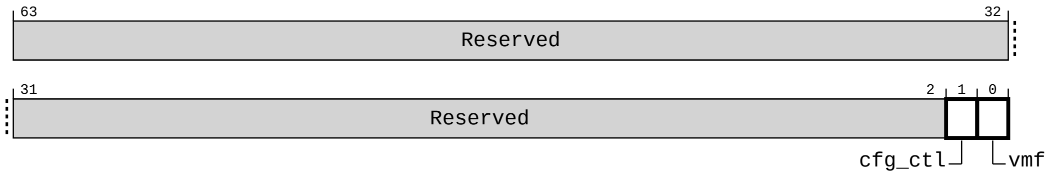


Table 5-122: por_dn_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	cfg_ctl	Allows Non-secure access to the Secure configuration control register, por_dn_cfg_ctl	RW	1'b0
[0]	vmf	Allows Non-secure access to Secure VMF registers	RW	1'b0

5.3.5.5 por_dn_cfg_ctl

Functions as the configuration control register for the DN.

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

Figure 5-107: por_dn_cfg_ctl

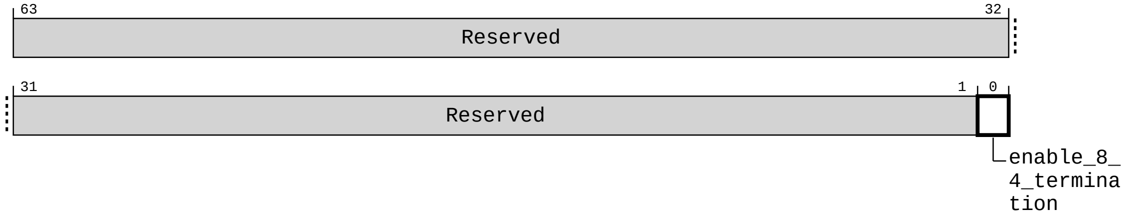


Table 5-123: por_dn_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	enable_8_4_termination	Enables termination of 8.4 DVMOps in DN	RW	1'b0

5.3.5.6 por_dn_aux_ctl

Functions as the auxiliary control register for the 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

Figure 5-108: por_dn_aux_ctl

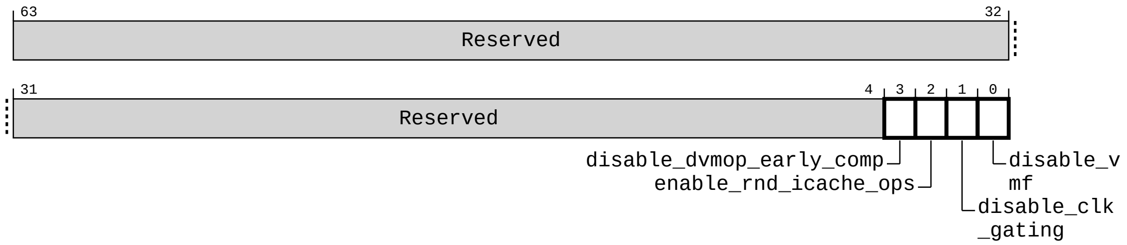


Table 5-124: por_dn_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	disable_dvmop_early_comp	Disables early Comp (CompDBID) for DVMOps	RW	1'b1
[2]	enable_rnd_icache_ops	Filters out BPI and VICI/PICI Snps to RN-Ds when set	RW	Configuration dependent
[1]	disable_clk_gating	Disables autonomous clock gating when set	RW	1'b0
[0]	disable_vmf	This bit is currently not supported. Software must not program this bit.	RW	Configuration dependent

5.3.5.7 por_dn_vmf0-15_ctrl

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Functions as the control register for VMID-based DVM snoop filtering. NOTE: This register has no effect when the disable_vmf field of the por_dn_aux_ctl register is set to 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC00 + (56 × #[0, 1, ... 15])

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

Figure 5-109: por_dn_vmf0-15_ctrl

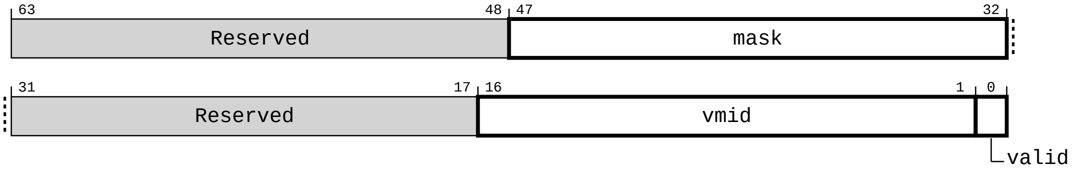


Table 5-125: por_dn_vmf0-15_ctrl attributes

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#{vmf_id}_ctrl.vmid. Then, the AND operator is performed on the mask and the VMID of the incoming request. 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 VMID of the incoming request is only compared with this VMID value if the VMID valid bit of the request is set. If the VMID of the request 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 disabled	RW	1'b0

5.3.5.8 por_dn_vmf0-15_rnf0

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Contains the logical RN-F bit vector[63:0] corresponding to por_dn_vmf#{vmf_id}_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 × #[0, 1, ... 15] + 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

Figure 5-110: por_dn_vmf0-15_rnf0

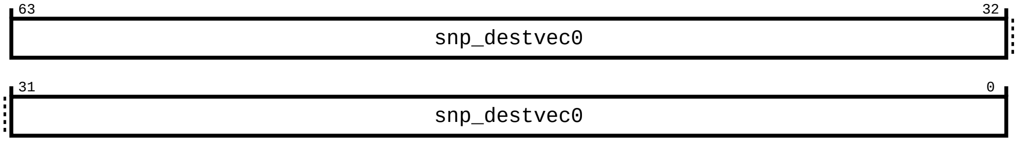


Table 5-126: por_dn_vmf0-15_rnf0 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec0	RN-F bit vector[63:0] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid	RW	64'b0

5.3.5.9 por_dn_vmf0-15_rnf1

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Contains the logical RN-F bit vector[127:64] corresponding to por_dn_vmf#{vmf_id}_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 × #[0, 1, ... 15] + 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

Figure 5-111: por_dn_vmf0-15_rnf1

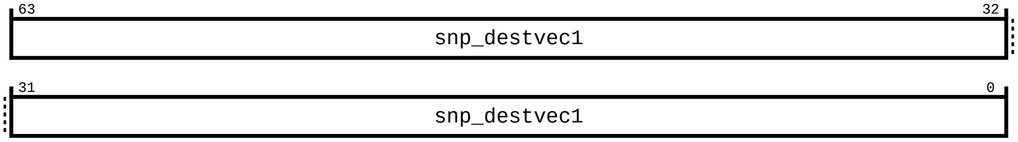


Table 5-127: por_dn_vmf0-15_rnf1 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec1	RN-F bit vector[127:64] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid	RW	64'b0

5.3.5.10 por_dn_vmf0-15_rnf2

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Contains the logical RN-F bit vector[191:128] corresponding to por_dn_vmf#{vmf_id}_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 × #[0, 1, ... 15] + 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

Figure 5-112: por_dn_vmf0-15_rnf2

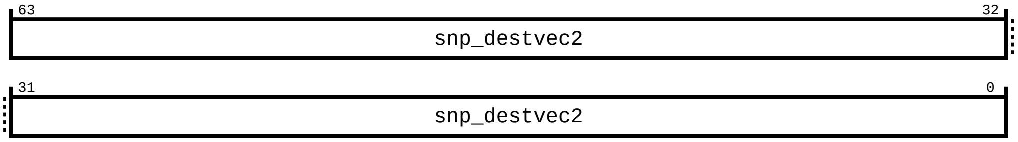


Table 5-128: por_dn_vmf0-15_rnf2 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec2	RN-F bit vector[191:128] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid	RW	64'b0

5.3.5.11 por_dn_vmf0-15_rnf3

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Contains the logical RN-F bit vector[255:192] corresponding to por_dn_vmf#{vmf_id}_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 × #[0, 1, ... 15] + 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

Figure 5-113: por_dn_vmf0-15_rnf3

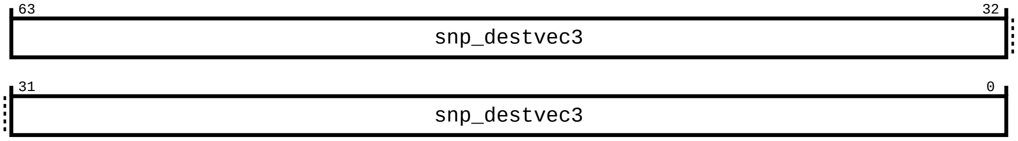


Table 5-129: por_dn_vmf0-15_rnf3 attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec3	RN-F bit vector[255:192] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid	RW	64'b0

5.3.5.12 por_dn_vmf0-15_rnd

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Contains the logical RN-D bit vector[63:0] corresponding to por_dn_vmf#{vmf_id}_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 × #[0, 1, ... 15] + 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

Figure 5-114: por_dn_vmf0-15_rnd

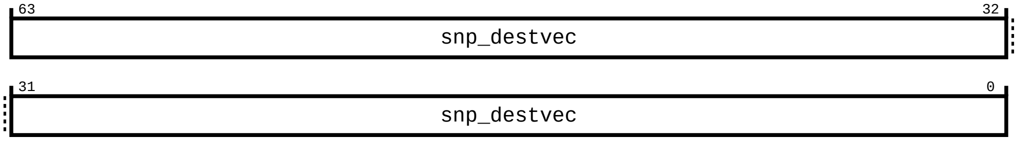


Table 5-130: por_dn_vmf0-15_rnd attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec	RN-D bit vector[63:0] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid	RW	64'b0

5.3.5.13 por_dn_vmf0-15_cxra

There are 16 iterations of this register, parameterized by the vmf_id from 0 to 15. Contains the logical CXRA bit vector[63:0] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid. Used for VMID-based DVM snoop filtering. NOTE: Not applicable and has no effect in a single-chip CMN-650 system.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC00 + (56 × #[0, 1, ... 15] + 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

Figure 5-115: por_dn_vmf0-15_cxra

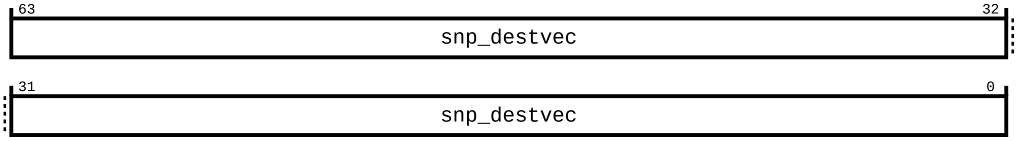


Table 5-131: por_dn_vmf0-15_cxra attributes

Bits	Name	Description	Type	Reset
[63:0]	snp_destvec	CXRA bit vector[63:0] corresponding to por_dn_vmf#{vmf_id}_ctrl.vmid	RW	64'b0

5.3.5.14 por_dn_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-116: por_dn_pmu_event_sel

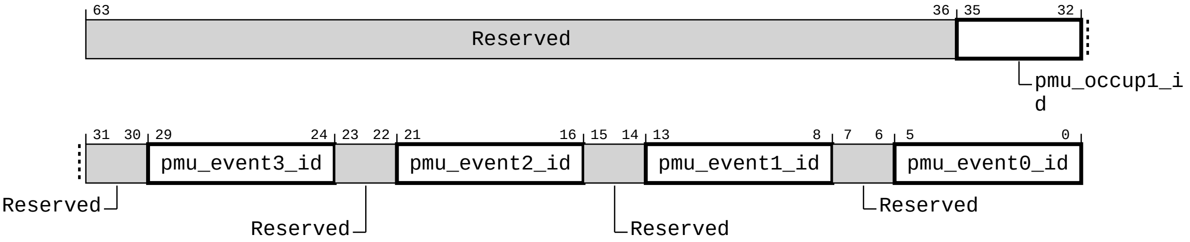


Table 5-132: por_dn_pmu_event_sel attributes

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: DVMOps 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 DVMOp requests 6'h02: Number of BPI DVMOp requests 6'h03: Number of PICI DVMOp requests 6'h04: Number of VICI DVMOp requests 6'h05: Number of DVM sync requests 6'h06: Number of DVMOp requests that were filtered using VMID filtering 6'h07: Number of BPI or PICI/VICI DVMOp requests to RN-Ds 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 credits 6'h0b: DVM tracker full counter 6'h0c: DVM tracker occupancy counter	RW	5'b0

5.3.6 CXHA register descriptions

This section lists the CXHA registers.

5.3.6.1 por_cxg_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

Figure 5-117: por_cxg_ha_node_info

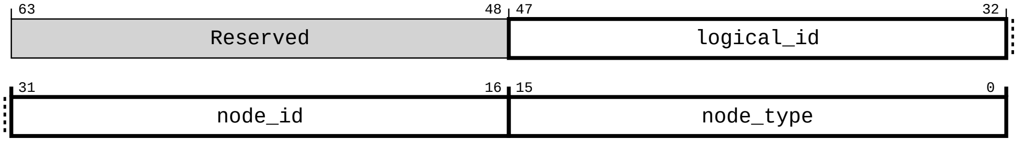


Table 5-133: por_cxg_ha_node_info attributes

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-650 node type identifier	RO	16'h0101

5.3.6.2 por_cxg_ha_id

Contains the CCIX-assigned Home Agent ID (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

Figure 5-118: por_cxg_ha_id

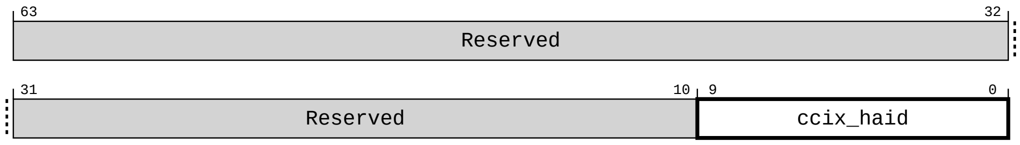


Table 5-134: por_cxg_ha_id attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:0]	ccix_haid	CCIX HAID	RW	10'h0

5.3.6.3 por_cxg_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

Figure 5-119: por_cxg_ha_child_info

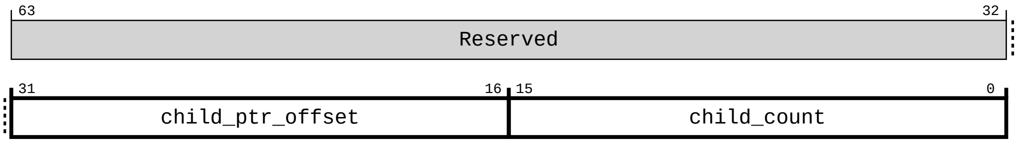


Table 5-135: por_cxg_ha_child_info attributes

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. This value is used in the discovery process.	RO	16'h0

5.3.6.4 por_cxg_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_cxg_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

Figure 5-120: por_cxg_ha_cfg_ctl

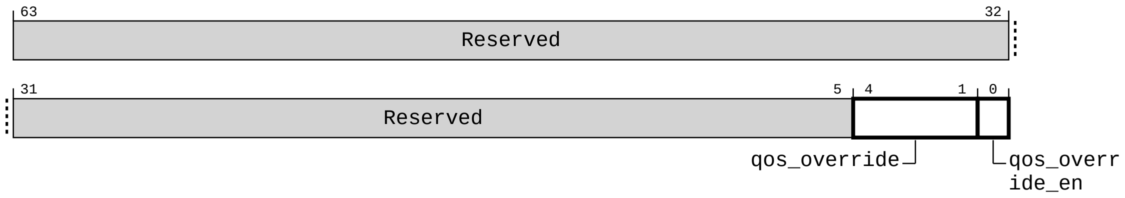


Table 5-136: por_cxg_ha_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:1]	qos_override	QoS override value	RW	4'b0
[0]	qos_override_en	QoS override enable. If set, the QoS value on the CHI side is driven from the qos_override value in this register.	RW	1'b0

5.3.6.5 por_cxg_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

Figure 5-121: por_cxg_ha_aux_ctl

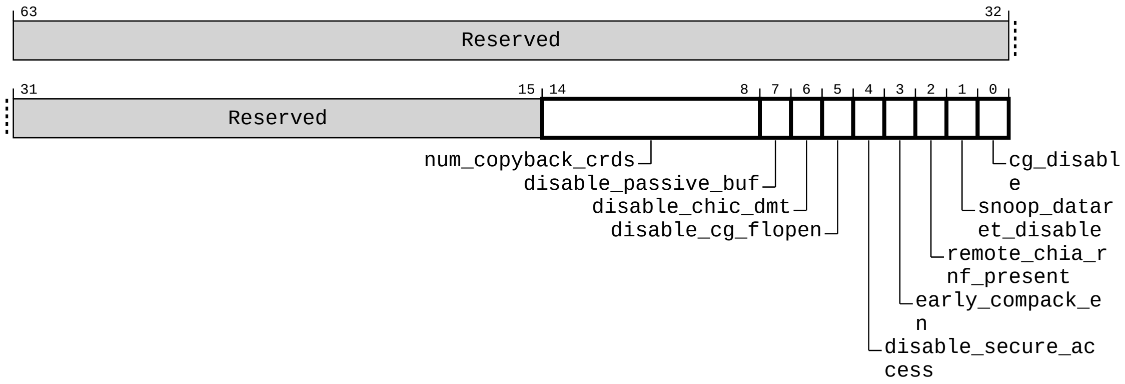


Table 5-137: por_cxg_ha_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:8]	num_copyback_crds	Controls the total number of request and data credits that are reserved for CopyBack Requests across all links Note: This value should be less than the write data buffer depth.	RW	7'h40
[7]	disable_passive_buf	If set, disables the passive buffer	RW	1'b0
[6]	disable_chic_dmt	If 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]	disable_secure_access	Converts all accesses to Non-secure	RW	1'b0
[3]	early_compack_en	Early CompAck enable. This bit enables sending early CompAck on CCIX for requests that require CompAck.	RW	1'b1
[2]	remote_chia_rnf_present	Indicates existence of CHI-A 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

5.3.6.6 por_cxg_ha_mpam_control

Controls MPAM override values on incoming CCIX requests in non-SMP mode.

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_cxg_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

Figure 5-122: por_cxg_ha_mpam_control

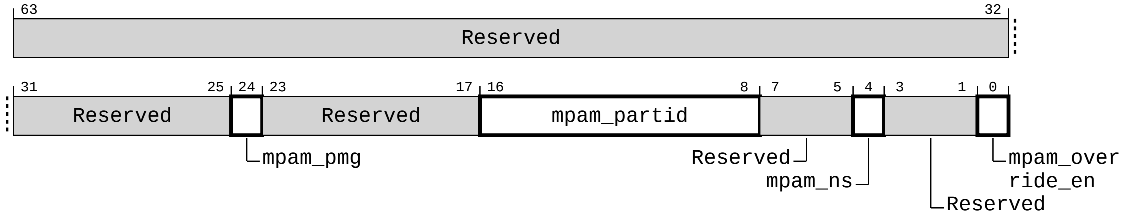


Table 5-138: por_cxg_ha_mpam_control attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	mpam_pmg	MPAM_PMG value	RW	1'b0
[23:17]	Reserved	Reserved	RO	-
[16:8]	mpam_partid	MPAM_PARTID value	RW	9'b0
[7:5]	Reserved	Reserved	RO	-
[4]	mpam_ns	MPAM_NS value	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	mpam_override_en	MPAM override enable. If set, the MPAM value on the CHI side is driven from the MPAM override value in this register. This value is only applicble in non-SMP mode.	RW	1'b0

5.3.6.7 por_cxg_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

Figure 5-123: por_cxg_ha_secure_register_groups_override

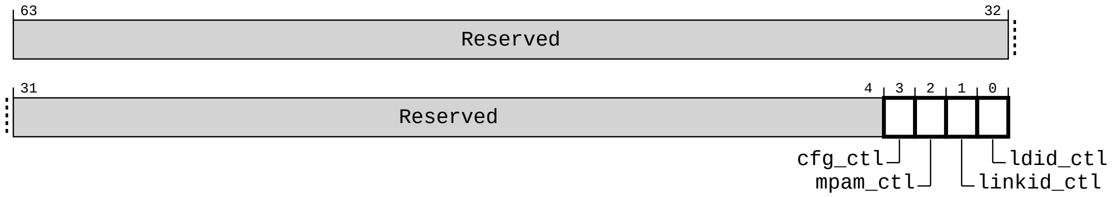


Table 5-139: por_cxg_ha_secure_register_groups_override attributes

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

5.3.6.8 por_cxg_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

Figure 5-124: por_cxg_ha_unit_info

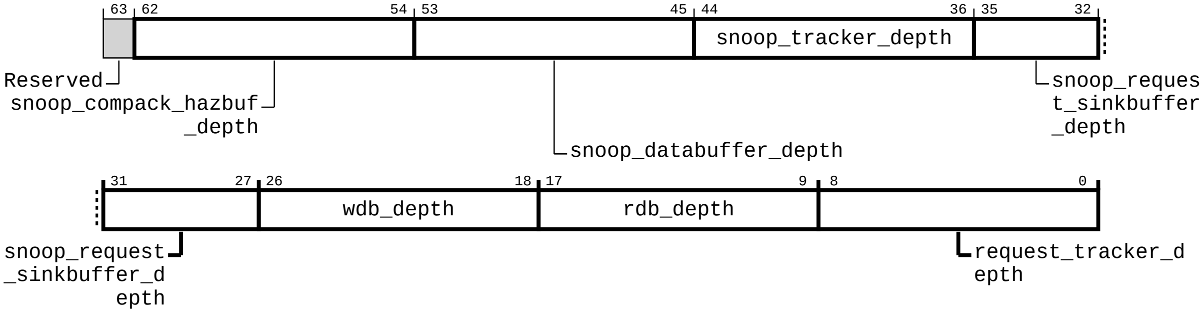


Table 5-140: por_cxg_ha_unit_info attributes

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. The depth determines the number of permitted outstanding SNP requests on CCIX.	RO	Configuration dependent
[35:27]	snoop_request_sinkbuffer_depth	Depth of snoop request sink buffer. The depth determines the number of CHI SNP requests that can be sunk by CXHA.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[26:18]	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

5.3.6.9 por_cxg_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

Figure 5-125: por_cxg_ha_unit_info2

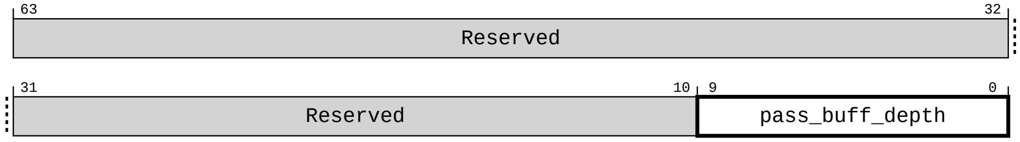


Table 5-141: por_cxg_ha_unit_info2 attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:0]	pass_buff_depth	Depth of the passive buffer	RO	Configuration dependent

5.3.6.10 por_cxg_ha_agentid_to_linkid_reg0

Specifies the mapping of Agent ID to Link ID for Agent IDs 0-7.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-126: por_cxg_ha_agentid_to_linkid_reg0

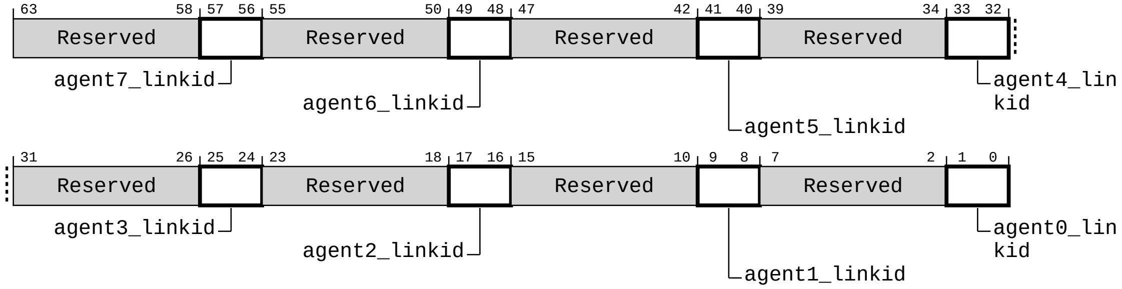


Table 5-142: por_cxg_ha_agentid_to_linkid_reg0 attributes

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

Bits	Name	Description	Type	Reset
[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

5.3.6.11 por_cxg_ha_agentid_to_linkid_reg1

Specifies the mapping of Agent ID to Link ID for Agent IDs 8-15.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-127: por_cxg_ha_agentid_to_linkid_reg1

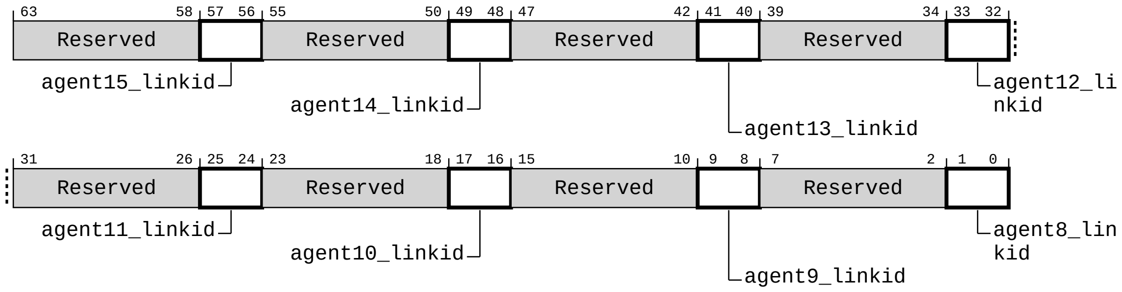


Table 5-143: por_cxg_ha_agentid_to_linkid_reg1 attributes

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	-
[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

5.3.6.12 por_cxg_ha_agentid_to_linkid_reg2

Specifies the mapping of Agent ID to Link ID for Agent IDs 16-23.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-128: por_cxg_ha_agentid_to_linkid_reg2

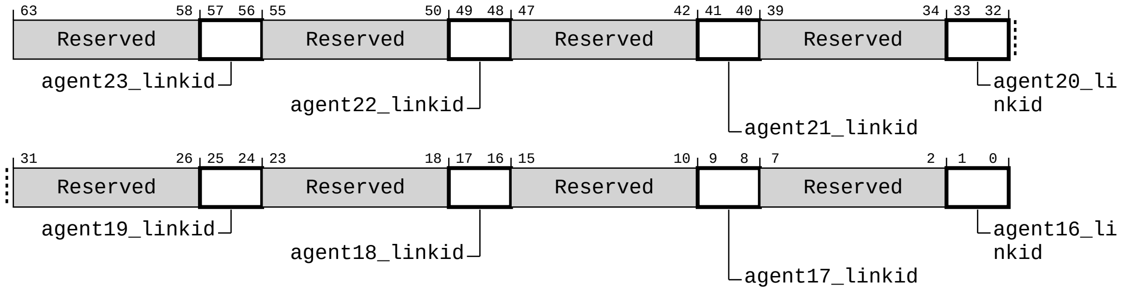


Table 5-144: por_cxg_ha_agentid_to_linkid_reg2 attributes

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

5.3.6.13 por_cxg_ha_agentid_to_linkid_reg3

Specifies the mapping of Agent ID to Link ID for Agent IDs 24-31.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-129: por_cxg_ha_agentid_to_linkid_reg3

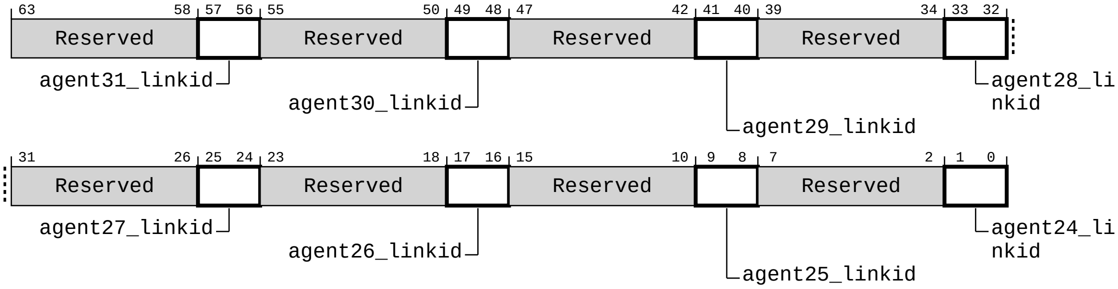


Table 5-145: por_cxg_ha_agentid_to_linkid_reg3 attributes

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	-

Bits	Name	Description	Type	Reset
[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

5.3.6.14 por_cxg_ha_agentid_to_linkid_reg4

Specifies the mapping of Agent ID to Link ID for Agent IDs 32-39.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-130: por_cxg_ha_agentid_to_linkid_reg4

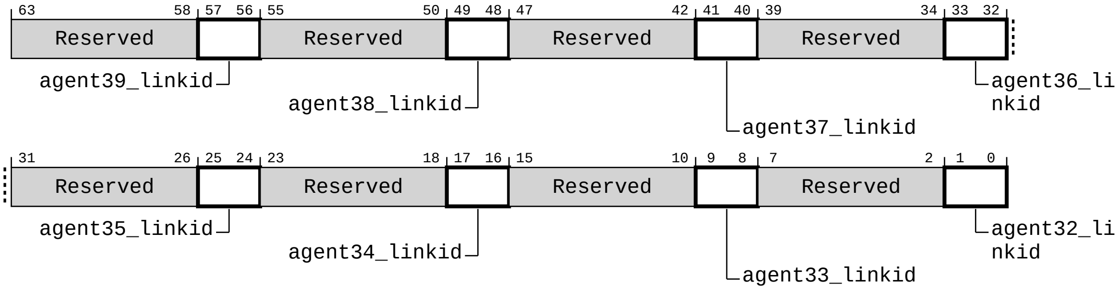


Table 5-146: por_cxg_ha_agentid_to_linkid_reg4 attributes

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	-
[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

5.3.6.15 por_cxg_ha_agentid_to_linkid_reg5

Specifies the mapping of Agent ID to Link ID for Agent IDs 40-47.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-131: por_cxg_ha_agentid_to_linkid_reg5

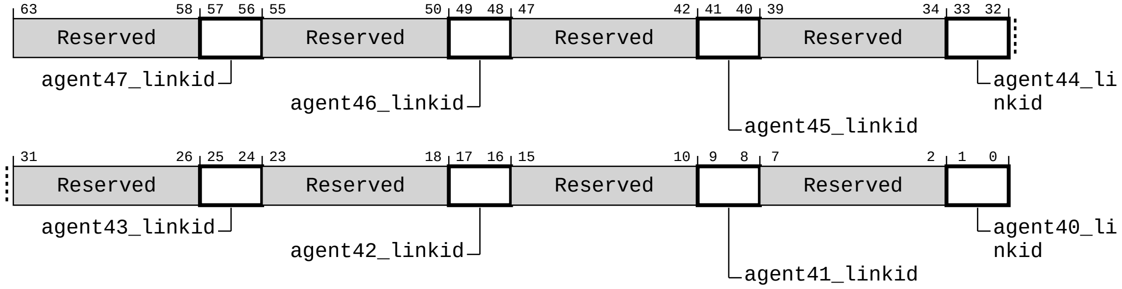


Table 5-147: por_cxg_ha_agentid_to_linkid_reg5 attributes

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

5.3.6.16 por_cxg_ha_agentid_to_linkid_reg6

Specifies the mapping of Agent ID to Link ID for Agent IDs 48-55.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-132: por_cxg_ha_agentid_to_linkid_reg6

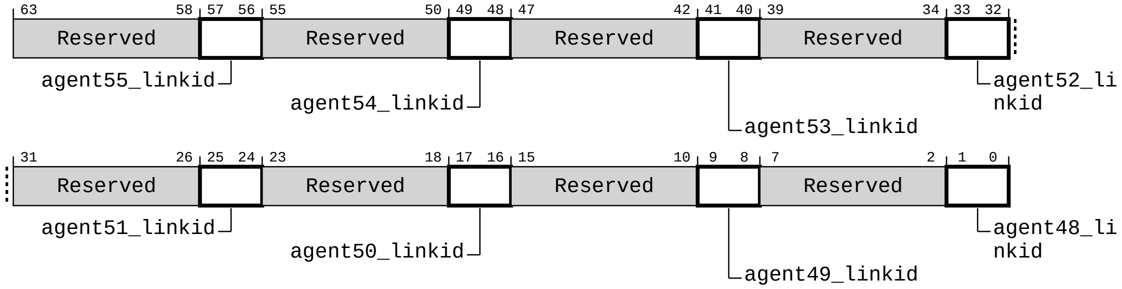


Table 5-148: por_cxg_ha_agentid_to_linkid_reg6 attributes

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	-

Bits	Name	Description	Type	Reset
[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

5.3.6.17 por_cxg_ha_agentid_to_linkid_reg7

Specifies the mapping of Agent ID to Link ID for Agent IDs 56-63.

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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-133: por_cxg_ha_agentid_to_linkid_reg7

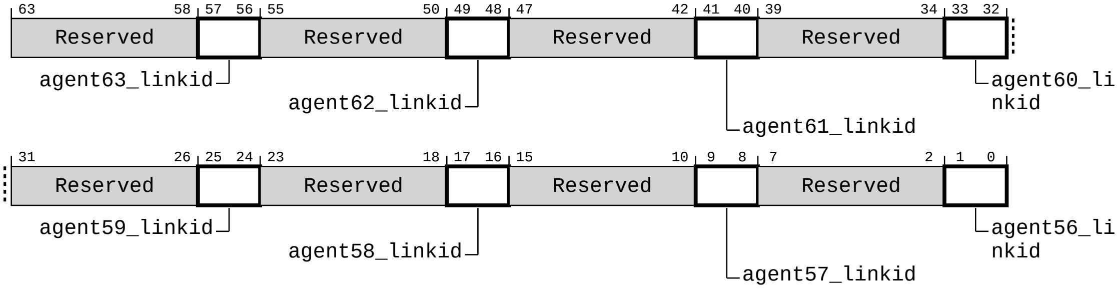


Table 5-149: por_cxg_ha_agentid_to_linkid_reg7 attributes

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	-
[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

5.3.6.18 por_cxg_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_cxg_ha_secure_register_groups_override.linkid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-134: por_cxg_ha_agentid_to_linkid_val

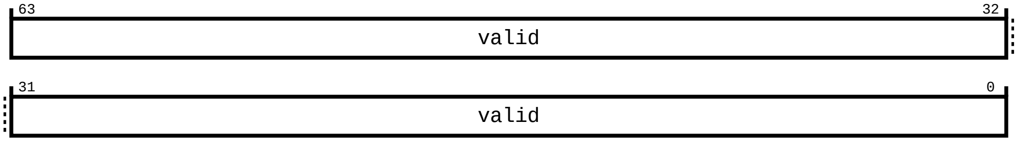


Table 5-150: por_cxg_ha_agentid_to_linkid_val attributes

Bits	Name	Description	Type	Reset
[63:0]	valid	Specifies whether the Link ID is valid. The bit number corresponds to the logical Agent ID number, from 0-63.	RW	63'h0

5.3.6.19 por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255

There are 255 iterations of this register, parameterized by the index from 0 to 255. Specifies the mapping of the expanded Request Agent ID (RAID) to the RN-F Logical Device ID (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 × #[0, 1, ... 255])

Type

RW

Reset value

See individual bit resets

Secure group override

por_cxg_ha_secure_register_groups_override.ldid_ctl

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-135: por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255

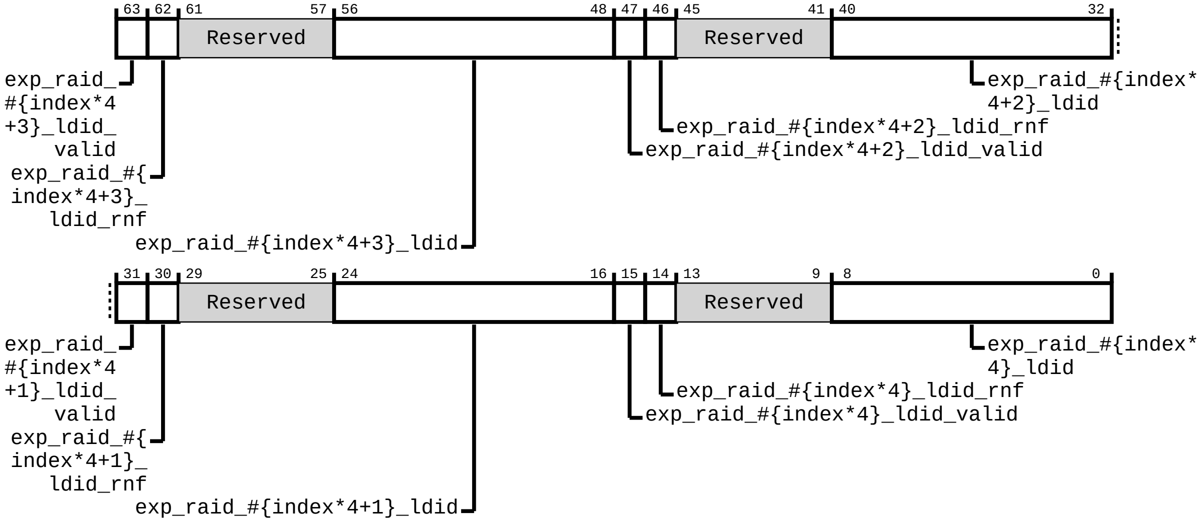

63
62
61
57
56
48
47
46
45
41
40
32
Reserved
Reserved
exp_raid_#{index*4+3}_ldid_valid
exp_raid_#{index*4+3}_ldid_rnf
exp_raid_#{index*4+3}_ldid
exp_raid_#{index*4+2}_ldid
exp_raid_#{index*4+2}_ldid_rnf
exp_raid_#{index*4+2}_ldid_valid
31
30
29
25
24
16
15
14
13
9
8
0
Reserved
Reserved
exp_raid_#{index*4+1}_ldid_valid
exp_raid_#{index*4+1}_ldid_rnf
exp_raid_#{index*4+1}_ldid
exp_raid_#{index*4}_ldid
exp_raid_#{index*4}_ldid_rnf
exp_raid_#{index*4}_ldid_valid

Table 5-151: por_cxg_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 the 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} corresponds to an 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 the 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} corresponds to an 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 the 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} corresponds to an 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 the LDID for expanded RAID #{index*4} is valid	RW	1'b0
[14]	exp_raid_#{index*4}_ldid_rnf	Specifies if expanded RAID #{index*4} corresponds to an RN-F	RW	1'b0
[13:9]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[8:0]	exp_raid_#{index*4}_ldid	Specifies the LDID for expanded RAID #{index*4}	RW	9'h0

5.3.6.20 por_cxg_ha_pmu_event_sel

Specifies the PMU event to be counted as a 6-bit ID.

The following encodings apply:

6'b000000	CXHA_PMU_EVENT_NULL
6'b100001	CXHA_PMU_EVENT_RDDATBYP
6'b100010	CXHA_PMU_EVENT_CHIRSP_UP_STALL
6'b100011	CXHA_PMU_EVENT_CHIDAT_UP_STALL
6'b100100	CXHA_PMU_EVENT_SNPPCRD_LNK0_STALL
6'b100101	CXHA_PMU_EVENT_SNPPCRD_LNK1_STALL
6'b100110	CXHA_PMU_EVENT_SNPPCRD_LNK2_STALL
6'b100111	CXHA_PMU_EVENT_REQTRK_OCC
6'b101000	CXHA_PMU_EVENT_RDB_OCC
6'b101001	CXHA_PMU_EVENT_RDBBYP_OCC
6'b101010	CXHA_PMU_EVENT_WDB_OCC
6'b101011	CXHA_PMU_EVENT_SNPTRK_OCC
6'b101100	CXHA_PMU_EVENT_SDB_OCC
6'b101101	CXHA_PMU_EVENT_SNPHAZ_OCC

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

Figure 5-136: por_cxg_ha_pmu_event_sel

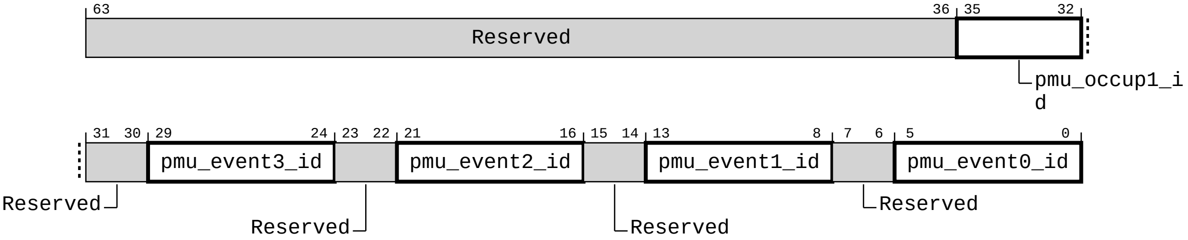


Table 5-152: por_cxg_ha_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:36]	Reserved	Reserved	RO	-
[35:32]	pmu_occup1_id	CXHA PMU occupancy event selector ID	RW	4'b0
[31:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	CXHA PMU event 3 ID	RW	6'b0
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	CXHA PMU event 2 ID	RW	6'b0
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	CXHA PMU event 1 ID	RW	6'b0
[7:6]	Reserved	Reserved	RO	-
[5:0]	pmu_event0_id	CXHA PMU event 0 ID	RW	6'b0

5.3.6.21 por_cxg_ha_cxprtcl_link0_ctl

Functions as the CXHA CCIX protocol link 0 control register. Works with the por_cxg_ha_cxprtcl_link0_status register.

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

Figure 5-137: por_cxg_ha_cxprtcl_link0_ctl

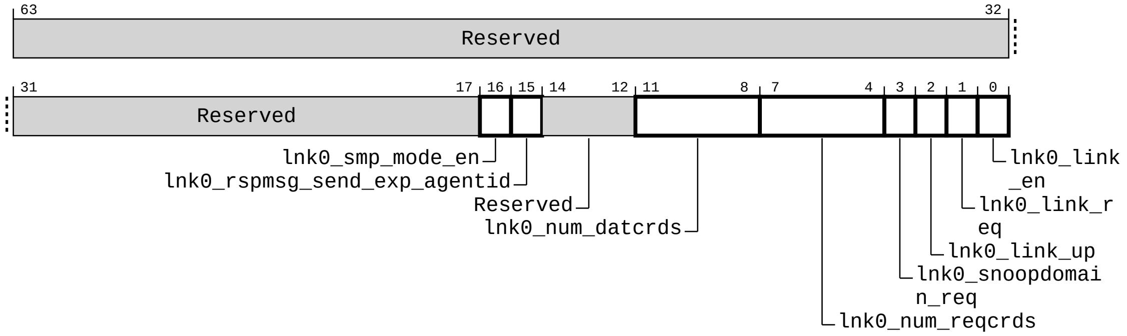


Table 5-153: por_cxg_ha_cxprtcl_link0_ctl attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	lnk0_smp_mode_en	If set, enables Symmetric Multiprocessor (SMP) mode for CCIX link 0	RW	Configuration dependent
[15]	lnk0_rspmsg_send_exp_agentid	If 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 that are 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 that are 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 the link status after polling the lnk<x>_link_ack and lnk<x>_link_down status in the remote agent: 1'b0: Link is not up. Software clears lnk<x>_link_up when lnk<x>_link_ack is clear and lnk<x>_link_down is set in both local and remote agents. The local agent stops responding to any protocol activity from the remote agent, including acceptance of protocol credits, when lnk<x>_link_up is clear. 1'b1: Link is up. Software sets lnk<x>_link_up when lnk<x>_link_ack is set and lnk<x>_link_down is clear in both local and remote agents. The local agent starts sending local protocol credits to the remote agent.	RW	1'b0
[1]	lnk0_link_req	Request link up or link down. Software writes this register bit to request a link up or link down in the local agent: 1'b0: Link down request Note: The local agent does not return remote protocol credits yet, because the remote agent may still be in the link up state. 1'b1: Link up request	RW	1'b0
[0]	lnk0_link_en	If set, enables CCIX link 0: 1'b0: Link is disabled 1'b1: Link is enabled	RW	1'b0

5.3.6.22 por_cxg_ha_cxprtcl_link0_status

Functions as the CXHA CCIX protocol link 0 status register. Works with the por_cxg_ha_cxprtcl_link0_ctl register.

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

Figure 5-138: por_cxg_ha_cxprtcl_link0_status

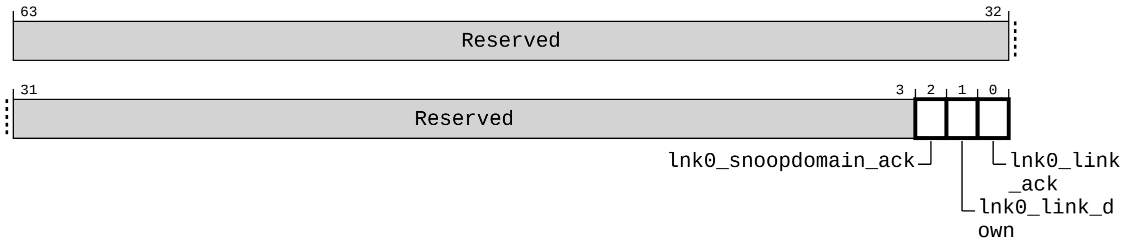


Table 5-154: por_cxg_ha_cxprtcl_link0_status attributes

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. The hardware updates this register bit to indicate link down status. 1'b0: Link is not down. The hardware clears lnk<x>_link_down when it receives a link up request. 1'b1: Link is down. The hardware sets lnk<x>_link_down 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]	lnk0_link_ack	Acknowledges link up or link down request. The hardware updates this register bit to acknowledge the software link request: 1'b0: Link down request acknowledged. The hardware clears lnk<x>_link_ack when it receives a link down request. The local agent stops sending protocol credits to the remote agent when lnk<x>_link_ack is clear. 1'b1: Link up request acknowledged. The hardware sets lnk<x>_link_ack when the local agent is ready to start accepting protocol credits from the remote agent. Note: The local agent must clear lnk<x>_link_down before setting lnk<x>_link_ack.	RO	1'b0

5.3.6.23 por_cxg_ha_cxprtcl_link1_ctl

Functions as the CXHA CCIX protocol link 1 control register. Works with the por_cxg_ha_cxprtcl_link1_status register.

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

Figure 5-139: por_cxg_ha_cxprtcl_link1_ctl

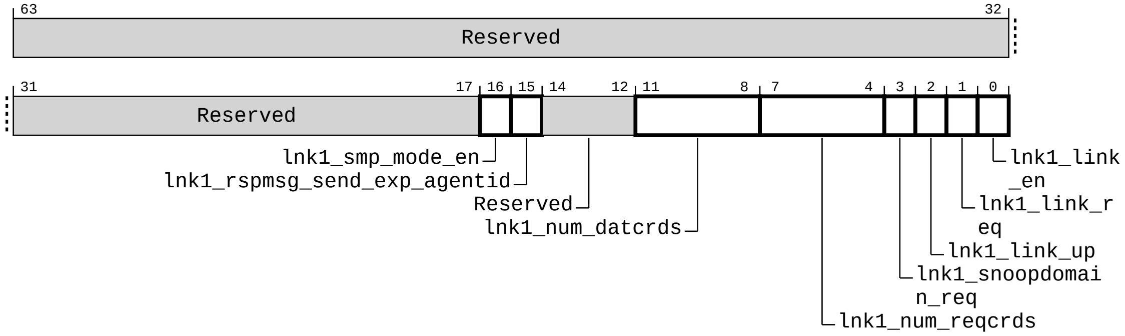


Table 5-155: por_cxg_ha_cxprtcl_link1_ctl attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	lnk1_smp_mode_en	If set, enables Symmetric Multiprocessor (SMP) mode for CCIX link 1	RW	Configuration dependent
[15]	lnk1_rspmsg_send_exp_agentid	If 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 that are assigned to Link 1: 4'h0: Total credits 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 that are 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

Bits	Name	Description	Type	Reset
[2]	lnk1_link_up	Link up status. Software writes this register bit to indicate link status after polling lnk<x>_link_ack and lnk<x>_link_down status in the remote agent: 1'b0: Link is not up. Software clears lnk<x>_link_up when lnk<x>_link_ack is clear and lnk<x>_link_down is set in both local and remote agents. The local agent stops responding to any protocol activity from the remote agent, including acceptance of protocol credits, when lnk<x>_link_up is clear. 1'b1: Link is up. Software sets lnk<x>_link_up when lnk<x>_link_ack is set and lnk<x>_link_down is clear in both local and remote agents. The local agent starts sending local protocol credits to the remote agent.	RW	1'b0
[1]	lnk1_link_req	Request link up or link down. Software writes this register bit to request a link up or link down in the local agent: 1'b0: Link down request Note: The local agent does not return remote protocol credits yet, because the remote agent may still be in link up state. 1'b1: Link up request	RW	1'b0
[0]	lnk1_link_en	If set, enables CCIX link 1: 1'b0: Link is disabled 1'b1: Link is enabled	RW	1'b0

5.3.6.24 por_cxg_ha_cxprtcl_link1_status

Functions as the CXHA CCIX protocol link 1 status register. Works with the por_cxg_ha_cxprtcl_link1_ctl register.

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

Figure 5-140: por_cxg_ha_cxprtcl_link1_status

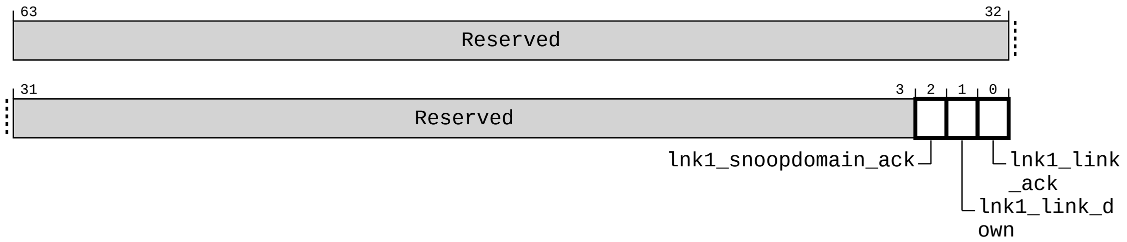


Table 5-156: por_cxg_ha_cxprtcl_link1_status attributes

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. The hardware updates this register bit to indicate link down status: 1'b0: Link is not down. The hardware clears lnk<x>_link_down when it receives a link up request. 1'b1: Link is down. The hardware sets lnk<x>_link_down 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	Acknowledges link up or link down request. The hardware updates this register bit to acknowledge the software link request: 1'b0: Link down request acknowledged. The hardware clears lnk<x>_link_ack when it receives a link down request. The local agent stops sending protocol credits to the remote agent when lnk<x>_link_ack is clear. 1'b1: Link up request acknowledged. The hardware sets lnk<x>_link_ack when the local agent is ready to start accepting protocol credits from the remote agent. Note: The local agent must clear lnk<x>_link_down before setting lnk<x>_link_ack.	RO	1'b0

5.3.6.25 por_cxg_ha_cxprtcl_link2_ctl

Functions as the CXHA CCIX protocol link 2 control register. Works with the por_cxg_ha_cxprtcl_link2_status register.

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

Figure 5-141: por_cxg_ha_cxprtcl_link2_ctl

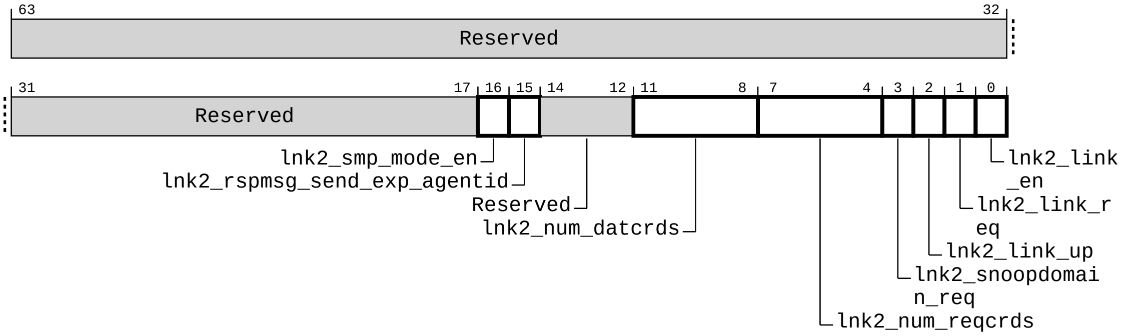


Table 5-157: por_cxg_ha_cxprtcl_link2_ctl attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	lnk2_smp_mode_en	If set, enables Symmetric Multiprocessor (SMP) mode for CCIX link 2	RW	Configuration dependent
[15]	lnk2_rspmsg_send_exp_agentid	If 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 that are 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

Bits	Name	Description	Type	Reset
[7:4]	lnk2_num_reqcrds	Controls the number of CCIX request credits that are 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 lnk<x>_link_ack and lnk<x>_link_down status in the remote agent: 1'b0: Link is not up. Software clears lnk<x>_link_up when lnk<x>_link_ack is clear and lnk<x>_link_down 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 lnk<x>_link_up is clear. 1'b1: Link is up. Software sets lnk<x>_link_up when lnk<x>_link_ack is set and lnk<x>_link_down is clear in both local and remote agents. The local agent starts sending local protocol credits to the remote agent.	RW	1'b0
[1]	lnk2_link_req	Request link up or link down request. Software writes this register bit to request a link up or link down in the local agent: 1'b0: Link down request Note: The local agent does not return remote protocol credits yet, because the remote agent may still be in lnk<x>_link_up state. 1'b1: Link up request	RW	1'b0
[0]	lnk2_link_en	If set, enables CCIX link 2: 1'b0: Link is disabled 1'b1: Link is enabled	RW	1'b0

5.3.6.26 por_cxg_ha_cxprtcl_link2_status

Functions as the CXHA CCIX protocol link 2 status register. Works with the por_cxg_ha_cxprtcl_link2_ctl register.

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

Figure 5-142: por_cxg_ha_cxprtcl_link2_status

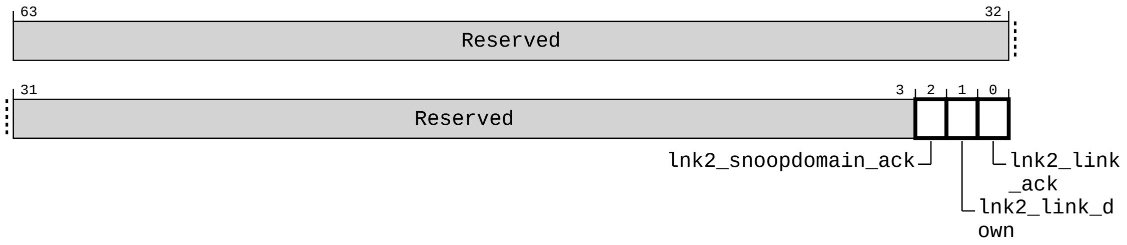


Table 5-158: por_cxg_ha_cxprtcl_link2_status attributes

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. The hardware updates this register bit to indicate link down status: 1'b0: Link is not down. The hardware clears lnk<x>_link_down when it receives a link up request. 1'b1: Link is down. The hardware sets lnk<x>_link_down 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	Acknowledges link up or link down. The hardware updates this register bit to acknowledge the software link request: 1'b0: Link down request acknowledged. The hardware clears lnk<x>_link_ack when it receives a link down request. The local agent stops sending protocol credits to the remote agent when lnk<x>_link_ack is clear. 1'b1: Link up request acknowledged. The hardware sets lnk<x>_link_ack when the local agent is ready to start accepting protocol credits from the remote agent. Note: The local agent must clear lnk<x>_link_down before setting lnk<x>_link_ack.	RO	1'b0

5.3.6.27 por_cxg_ha_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-143: por_cxg_ha_errfr

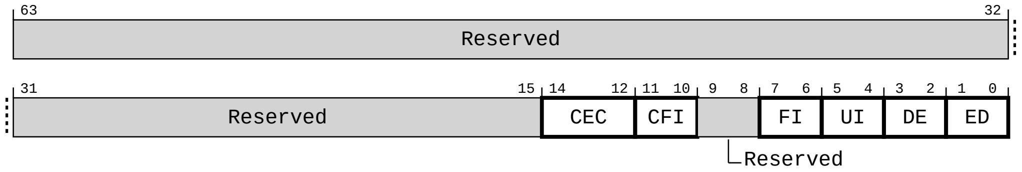


Table 5-159: por_cxg_ha_errfr attributes

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_cxg_ha_errmisc[39:32] 3'b100: Implements 16-bit error counter in por_cxg_ha_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

5.3.6.28 por_cxg_ha_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-144: por_cxg_ha_errctlr

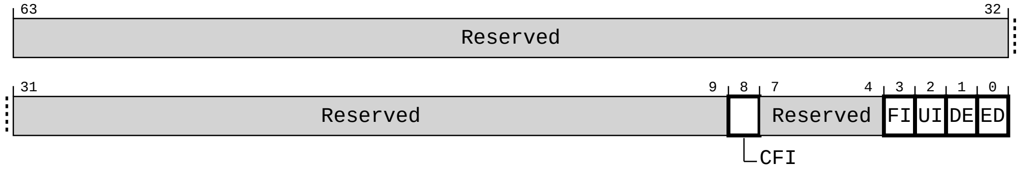


Table 5-160: por_cxg_ha_errctlr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_cxg_ha_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_cxg_ha_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_cxg_ha_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_cxg_ha_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_cxg_ha_errfr.ED	RW	1'b0

5.3.6.29 por_cxg_ha_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-145: por_cxg_ha_errstatus

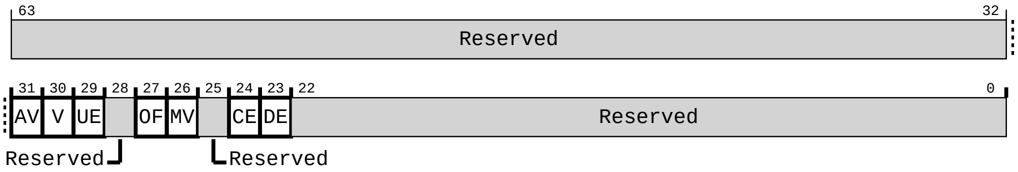


Table 5-161: por_cxg_ha_errstatus attributes

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_cxg_ha_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 slave 1'b0: No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow. This bit is 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, or CE fields	W1C	1'b0
[26]	MV	por_cxg_ha_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	-

5.3.6.30 por_cxg_ha_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-146: por_cxg_ha_erraddr

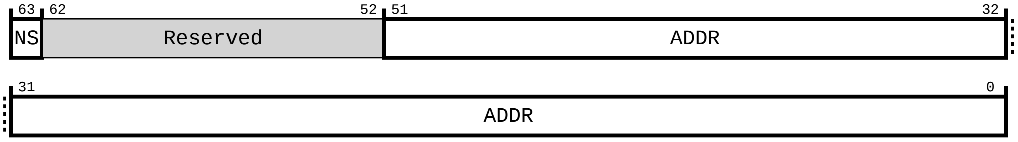


Table 5-162: por_cxg_ha_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction: 1'b1: Non-secure transaction 1'b0: Secure transaction CONSTRAINT: por_cxg_ha_erraddr.NS is redundant. Because 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

5.3.6.31 por_cxg_ha_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-147: por_cxg_ha_errmisc

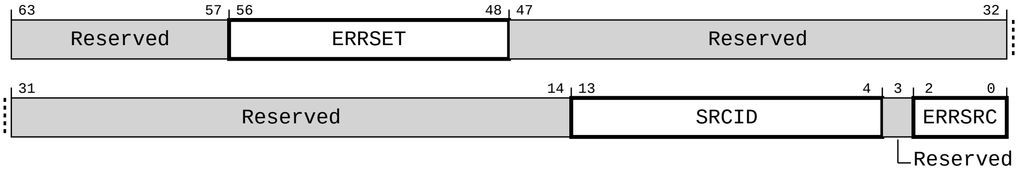


Table 5-163: por_cxg_ha_errmisc attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ERRSET	RAM entry set address for parity error	RW	9'b0
[47:14]	Reserved	Reserved	RO	-
[13:4]	SRCID	CCIX Request Agent ID (RAID) of the requestor or the snoop target	RW	10'b0
[3]	Reserved	Reserved	RO	-
[2:0]	ERRSRC	Source of the parity error: 3'b000: Read data buffer 0 3'b001: Read data buffer 1 3'b010: Write data buffer 0 3'b011: Write data buffer 1 3'b100: Passive buffer	RW	3'b000

5.3.6.32 por_cxg_ha_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

Figure 5-148: por_cxg_ha_errfr_NS

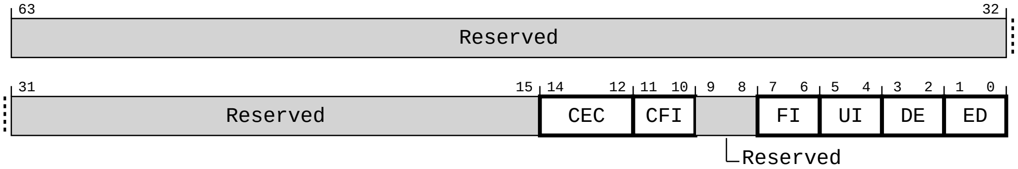


Table 5-164: por_cxg_ha_errfr_NS attributes

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_cxg_ha_errmisc[39:32] 3'b100: Implements 16-bit error counter in por_cxg_ha_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

5.3.6.33 por_cxg_ha_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection and 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

Figure 5-149: por_cxg_ha_errctlr_NS

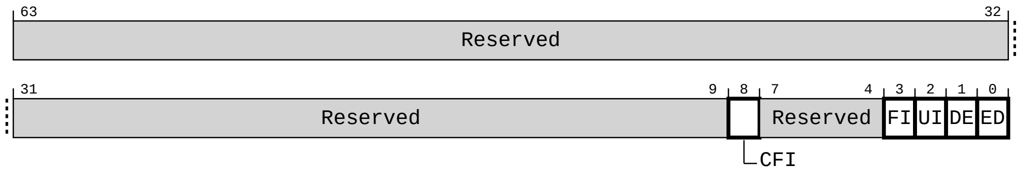


Table 5-165: por_cxg_ha_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_cxg_ha_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_cxg_ha_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_cxg_ha_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_cxg_ha_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_cxg_ha_errfr.ED	RW	1'b0

5.3.6.34 por_cxg_ha_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

Figure 5-150: por_cxg_ha_errstatus_NS

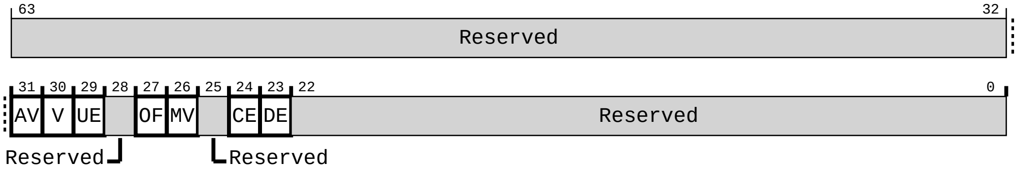


Table 5-166: por_cxg_ha_errstatus_NS attributes

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_cxg_ha_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 slave 1'b0: No uncorrected errors detected	W1C	1'b0
[28]	Reserved	Reserved	RO	-
[27]	OF	Overflow. This bit is 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_cxg_ha_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	-

5.3.6.35 por_cxg_ha_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

Figure 5-151: por_cxg_ha_erraddr_NS

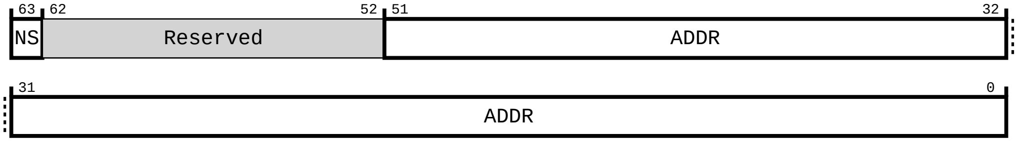


Table 5-167: por_cxg_ha_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction: 1'b1: Non-secure transaction 1'b0: Secure transaction CONSTRAINT: por_cxg_ha_erraddr.NS is redundant. Because 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

5.3.6.36 por_cxg_ha_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

Figure 5-152: por_cxg_ha_errmisc_NS

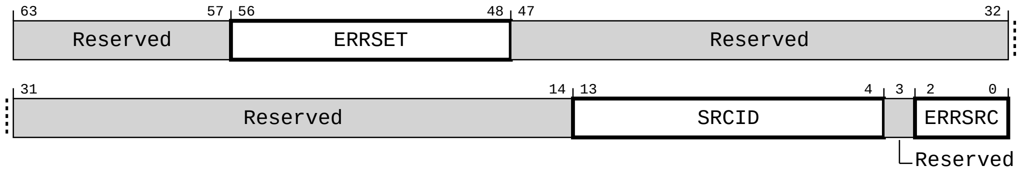


Table 5-168: por_cxg_ha_errmisc_NS attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:48]	ERRSET	RAM entry set address for parity error	RW	9'b0
[47:14]	Reserved	Reserved	RO	-
[13:4]	SRCID	CCIX Request Agent ID (RAID) of the requestor or the snoop target	RW	10'b0
[3]	Reserved	Reserved	RO	-
[2:0]	ERRSRC	Source of the parity error: 3'b000: Read data buffer 0 3'b001: Read data buffer 1 3'b010: Write data buffer 0 3'b011: Write data buffer 1 3'b100: Passive buffer	RW	3'b000

5.3.7 RN SAM register descriptions

This section lists the RN SAM registers.

5.3.7.1 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

Figure 5-153: por_rnsam_node_info

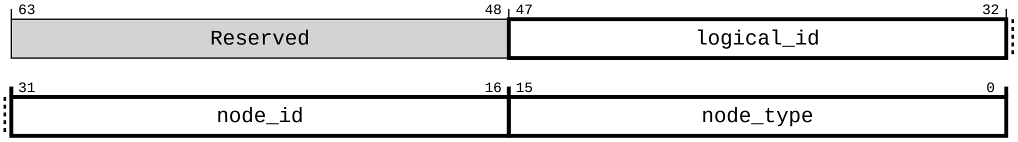


Table 5-169: por_rnsam_node_info attributes

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-650 node type identifier	RO	16'h000F

5.3.7.2 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

Figure 5-154: por_rnsam_child_info

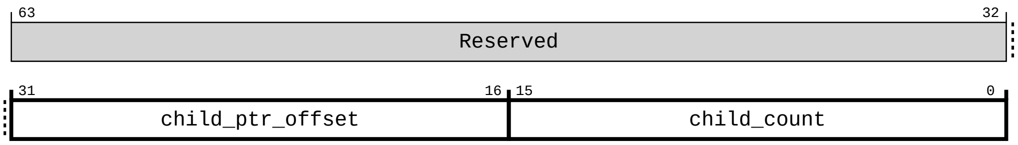


Table 5-170: por_rnsam_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.7.3 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

Figure 5-155: por_rnsam_secure_register_groups_override

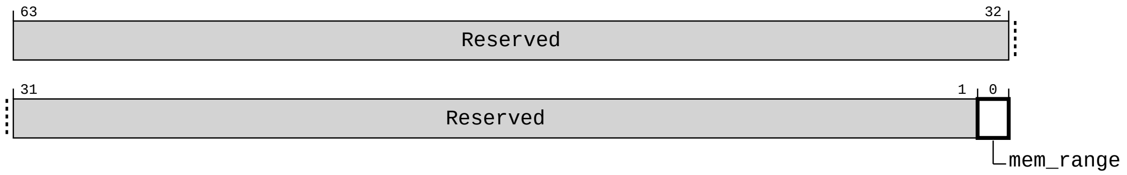


Table 5-171: por_rnsam_secure_register_groups_override attributes

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

5.3.7.4 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

Figure 5-156: por_rnsam_unit_info

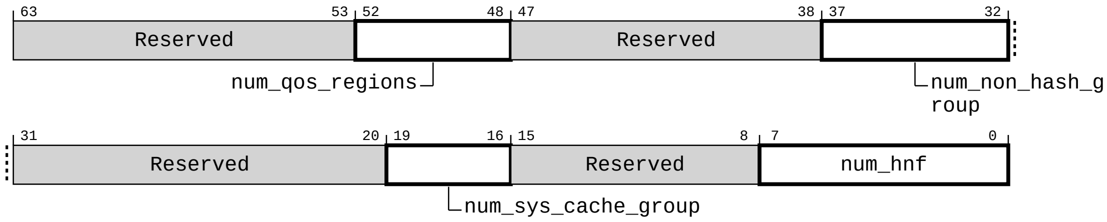


Table 5-172: por_rnsam_unit_info attributes

Bits	Name	Description	Type	Reset
[63:53]	Reserved	Reserved	RO	-
[52:48]	num_qos_regions	Number of QoS override regions supported	RO	Configuration dependent
[47:38]	Reserved	Reserved	RO	-
[37:32]	num_non_hash_group	Number of non-hashed groups supported	RO	Configuration dependent
[31:20]	Reserved	Reserved	RO	-
[19:16]	num_sys_cache_group	Number of System Cache Groups (SCGs) supported	RO	Configuration dependent
[15:8]	Reserved	Reserved	RO	-
[7:0]	num_hnf	Number of hashed targets supported	RO	Configuration dependent

5.3.7.5 non_hash_mem_region_reg0-19

There are 20 iterations of this register, parameterized by the index from 0 to 19. Configures non-hashed memory regions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC00 + (8 × #[0, 1, ... 19])

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

Figure 5-157: non_hash_mem_region_reg0-19

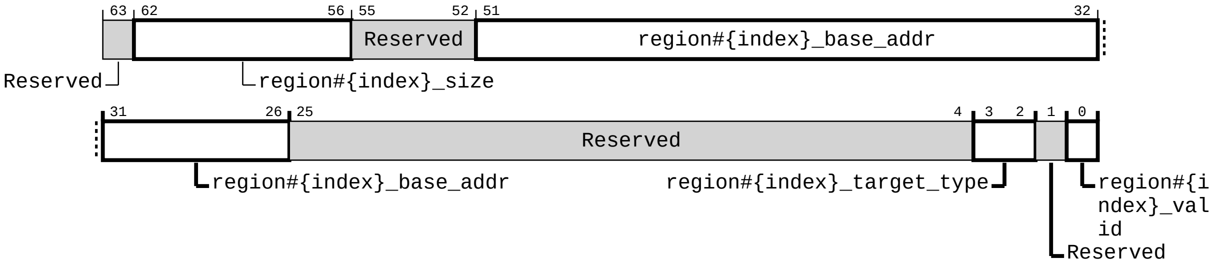


Table 5-173: non_hash_mem_region_reg0-19 attributes

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	-
[51:26]	region#{index}_base_addr	Bits[51:26] of base address of the memory region. CONSTRAINT: Must be an integer multiple of region #{index} size	RW	26'b00000000000000000000000000
[25:4]	Reserved	Reserved	RO	-
[3:2]	region#{index}_target_type	Indicates node type: 2'b00: HN-F 2'b01: HN-I 2'b10: CXRA 2'b11: Reserved CONSTRAINT: Only applicable for RN-I.	RW	2'b00
[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

5.3.7.6 non_hash_tgt_nodeid0-4

There are 5 iterations of this register, parameterized by the index from 0 to 4. Configures non-hashed target node IDs #{4*index + 0} to #{4*index + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD80 + (8 × #[0, 1, ... 4])

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

Figure 5-158: non_hash_tgt_nodeid0-4

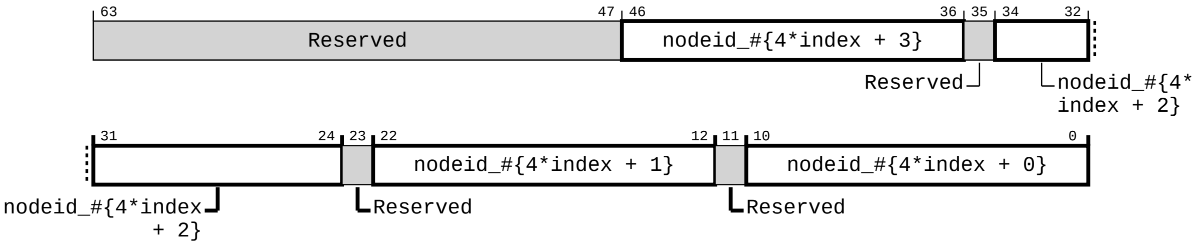


Table 5-174: non_hash_tgt_nodeid0-4 attributes

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	-

Bits	Name	Description	Type	Reset
[10:0]	nodeid_#{4*index + 0}	Non-hashed target node ID #{4*index + 0}	RW	11'b00000000000

5.3.7.7 sys_cache_grp_region0-3

There are 4 iterations of this register, parameterized by the index from 0 to 3. Configures hashed memory regions.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE00 + (8 × #[0, 1, ... 3])

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

Figure 5-159: sys_cache_grp_region0-3

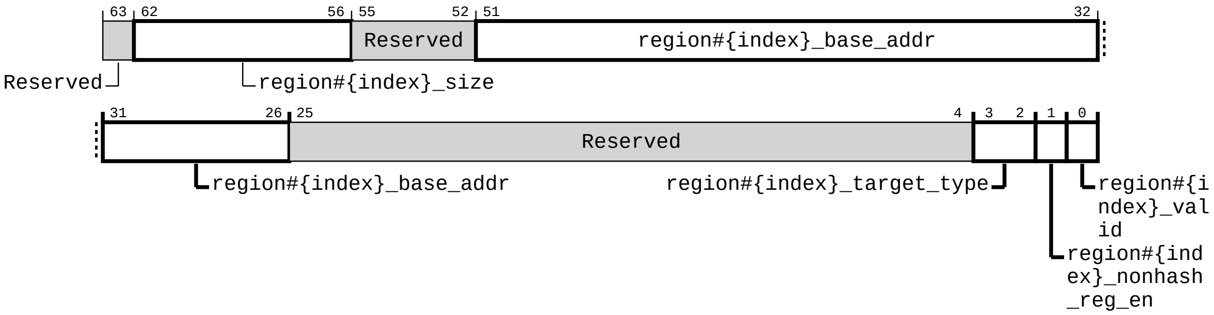


Table 5-175: sys_cache_grp_region0-3 attributes

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	-
[51:26]	region#{index}_base_addr	Bits[51:26] of base address of the memory region. CONSTRAINT: Must be an integer multiple of region #{index} size.	RW	26'b00000000000000000000000000
[25:4]	Reserved	Reserved	RO	-
[3:2]	region#{index}_target_type	Indicates node type: 2'b00: HN-F 2'b01: HN-I 2'b10: CXRA 2'b11: Reserved CONSTRAINT: Only applicable for RN-I	RW	2'b00
[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

5.3.7.8 sys_cache_grp_secondary_reg0-3

There are 4 iterations of this register, parameterized by the index from 0 to 3. Configures secondary hashed memory regions

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE40 + (8 × #[0, 1, ... 3])

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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-160: sys_cache_grp_secondary_reg0-3

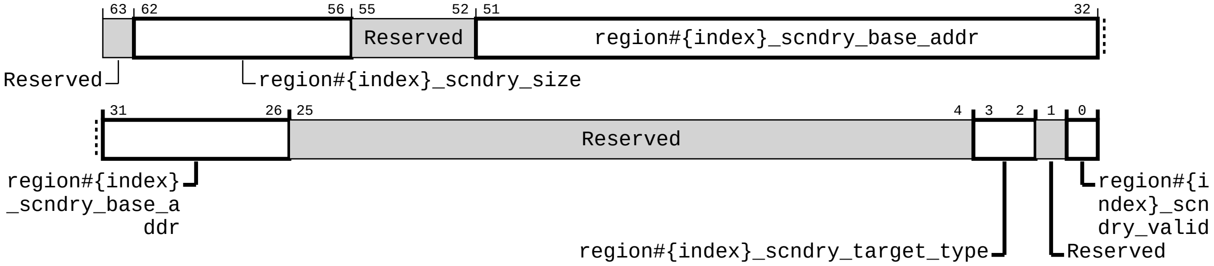

63
62
56
55
52
51
32
Reserved
region#{index}_scndry_base_addr
Reserved
region#{index}_scndry_size
31
26
25
4
3
2
1
0
Reserved
region#{index}_scndry_base_addr
region#{index}_scndry_valid
region#{index}_scndry_target_type
Reserved

Table 5-176: sys_cache_grp_secondary_reg0-3 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[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	-
[51:26]	region#{index}_scndry_base_addr	Bits[51:26] of secondary memory region base address. CONSTRAINT: Must be an integer multiple of region #{index} size	RW	26'b00000000000000000000000000
[25:4]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[3:2]	region#{index}_scndry_target_type	Indicates secondary node type: 2'b00: HN-F 2'b01: HN-I 2'b10: CXRA 2'b11: Reserved CONSTRAINT: Only applicable for RN-I.	RW	2'b00
[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

5.3.7.9 rnsam_hash_addr_mask_reg

Configures the address mask that is applied before hashing the address bits during RN SAM lookup.

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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-161: rnsam_hash_addr_mask_reg

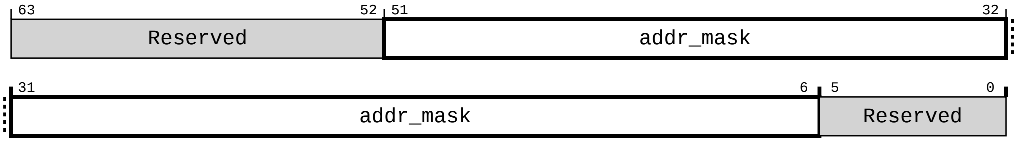


Table 5-177: rnsam_hash_addr_mask_reg attributes

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	-

5.3.7.10 rnsam_region_cmp_addr_mask_reg

Configures the address mask that is applied before region comparison during RN SAM lookup.

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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-162: rnsam_region_cmp_addr_mask_reg

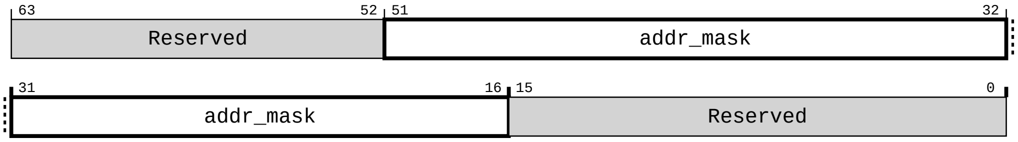


Table 5-178: rnsam_region_cmp_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	addr_mask	Address mask applied before memory region comparison	RW	36'hFFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

5.3.7.11 sys_cache_group_hn_count

Indicates number of HN-Fs in System Cache Groups (SCGs) 0-3.

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

Figure 5-163: sys_cache_group_hn_count

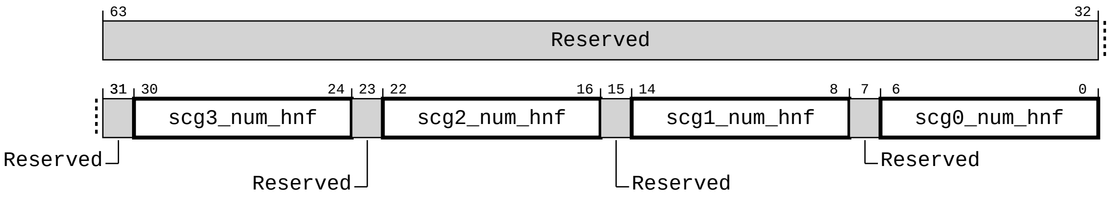


Table 5-179: sys_cache_group_hn_count attributes

Bits	Name	Description	Type	Reset
[63:31]	Reserved	Reserved	RO	-
[30:24]	scg3_num_hnf	HN-F count for SCG 3	RW	7'b00000
[23]	Reserved	Reserved	RO	-
[22:16]	scg2_num_hnf	HN-F count for SCG 2	RW	7'b00000
[15]	Reserved	Reserved	RO	-
[14:8]	scg1_num_hnf	HN-F count for SCG 1	RW	7'b00000
[7]	Reserved	Reserved	RO	-
[6:0]	scg0_num_hnf	HN-F count for SCG 0	RW	7'b00000

5.3.7.12 sys_cache_grp_sn_attr

Configures attributes for SN node IDs for System Cache Groups (SCGs).

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

Figure 5-164: sys_cache_grp_sn_attr

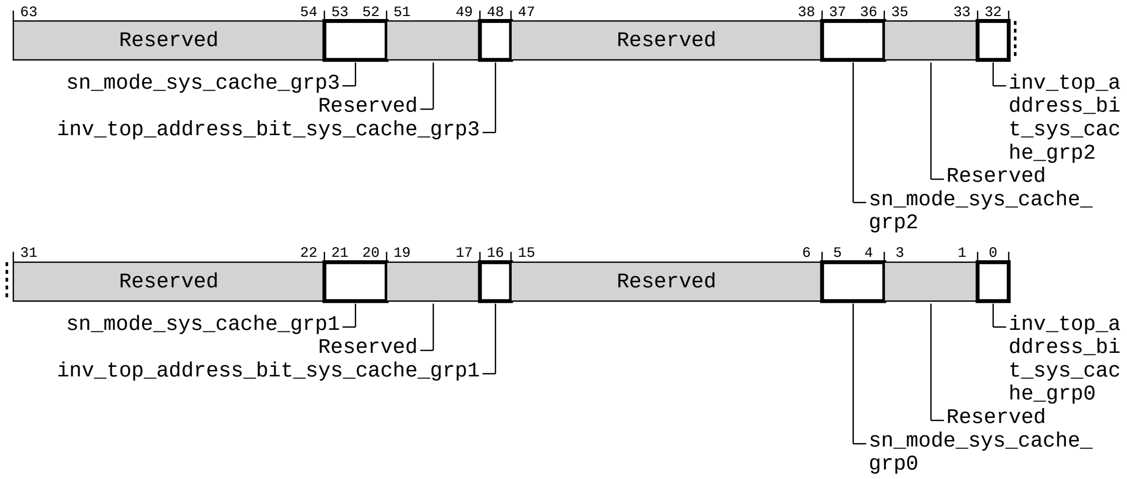


Table 5-180: sys_cache_grp_sn_attr attributes

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	-
[53:52]	sn_mode_sys_cache_grp3	SN selection mode: 2'b00: 1-SN mode (SN0) 2'b01: 3-SN mode (SN0, SN1, SN2) 2'b10: 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) 2'b11: 5-SN mode (SN0, SN1, SN2, SN3, SN4)	RW	2'b0
[51:49]	Reserved	Reserved	RO	-
[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 use this setting when the address map does not have unique address bit combinations.	RW	1'h0
[47:38]	Reserved	Reserved	RO	-
[37:36]	sn_mode_sys_cache_grp2	SN selection mode: 2'b00: 1-SN mode (SN0) 2'b01: 3-SN mode (SN0, SN1, SN2) 2'b10: 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) 2'b11: 5-SN mode (SN0, SN1, SN2, SN3, SN4)	RW	2'b00
[35:33]	Reserved	Reserved	RO	-
[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 use this setting when the address map does not have unique address bit combinations.	RW	1'h0

Bits	Name	Description	Type	Reset
[31:22]	Reserved	Reserved	RO	-
[21:20]	sn_mode_sys_cache_grp1	SN selection mode: 2'b00: 1-SN mode (SN0) 2'b01: 3-SN mode (SN0, SN1, SN2) 2'b10: 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) 2'b11: 5-SN mode (SN0, SN1, SN2, SN3, SN4)	RW	2'b0
[19:17]	Reserved	Reserved	RO	-
[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 use this setting when the address map does not have unique address bit combinations.	RW	1'h0
[15:6]	Reserved	Reserved	RO	-
[5:4]	sn_mode_sys_cache_grp0	SN selection mode: 2'b00: 1-SN mode (SN0) 2'b01: 3-SN mode (SN0, SN1, SN2) 2'b10: 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) 2'b11: 5-SN mode (SN0, SN1, SN2, SN3, SN4)	RW	2'b0
[3:1]	Reserved	Reserved	RO	-
[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 use this setting when the address map does not have unique address bit combinations.	RW	1'h0

5.3.7.13 sys_cache_grp_nonhash_nodeid

Configures non-hashed node IDs for System Cache Groups (SCGs) 1-3.

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

Figure 5-165: sys_cache_grp_nonhash_nodeid

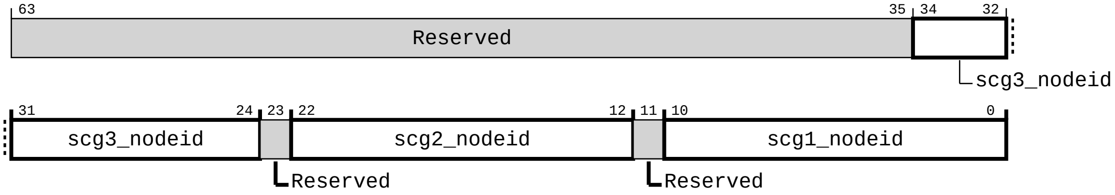


Table 5-181: sys_cache_grp_nonhash_nodeid attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34:24]	scg3_nodeid	Non-hashed node ID for SCG 3	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	scg2_nodeid	Non-hashed node ID for SCG 2	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	scg1_nodeid	Non-hashed node ID for SCG 1	RW	11'b00000000000

5.3.7.14 sys_cache_grp_hn_nodeid_reg0-15

There are 16 iterations of this register, parameterized by the index from 0 to 15. Configures hashed node IDs for System Cache Groups (SCGs). Controls target HN node IDs #{index*4 + 0} to #{index*4 + 3}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF00 + (8 × #[0, 1, ... 15])

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

Figure 5-166: sys_cache_grp_hn_nodeid_reg0-15

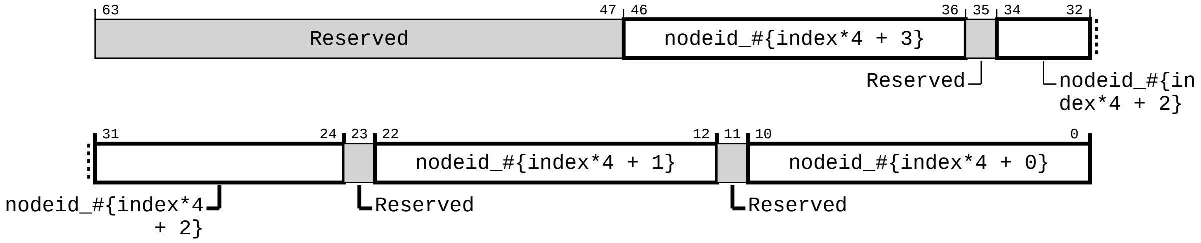


Table 5-182: sys_cache_grp_hn_nodeid_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:36]	nodeid_#{index*4 + 3}	Hashed target node ID #{index*4 + 3}	RW	11'b00000000000
[35]	Reserved	Reserved	RO	-
[34:24]	nodeid_#{index*4 + 2}	Hashed target node ID #{index*4 + 2}	RW	11'b00000000000
[23]	Reserved	Reserved	RO	-
[22:12]	nodeid_#{index*4 + 1}	Hashed target node ID #{index*4 + 1}	RW	11'b00000000000
[11]	Reserved	Reserved	RO	-
[10:0]	nodeid_#{index*4 + 0}	Hashed target node ID #{index*4 + 0}	RW	11'b00000000000

5.3.7.15 sys_cache_grp_sn_nodeid_reg0-15

There are 16 iterations of this register, parameterized by the index from 0 to 15. Configures hashed node IDs for System Cache Groups (SCGs). 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'h1000 + (8 × #[0, 1, ... 15])

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

Figure 5-167: sys_cache_grp_sn_nodeid_reg0-15

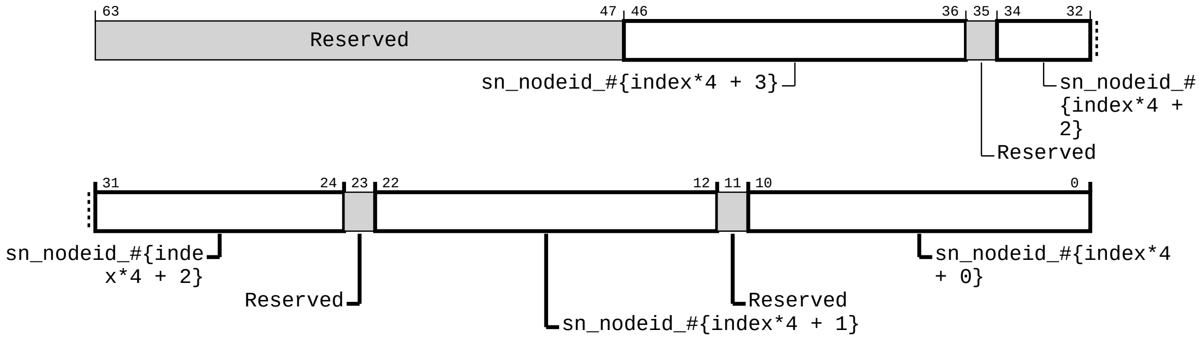


Table 5-183: sys_cache_grp_sn_nodeid_reg0-15 attributes

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 + 0}	Hashed target SN node ID #{index*4 + 0}	RW	11'b00000000000

5.3.7.16 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

Figure 5-168: rnsam_status

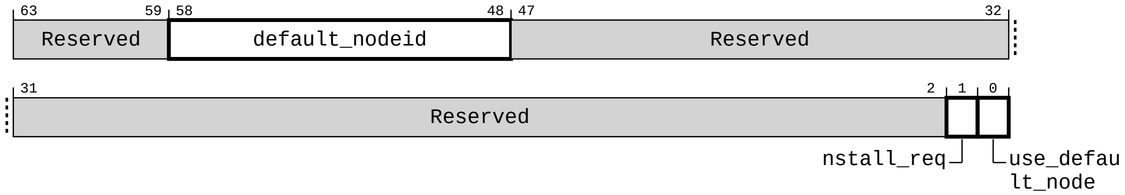


Table 5-184: rnsam_status attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	default_nodeid	Default Node ID	RW	Configuration dependent
[47:2]	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

5.3.7.17 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

Figure 5-169: gic_mem_region_reg

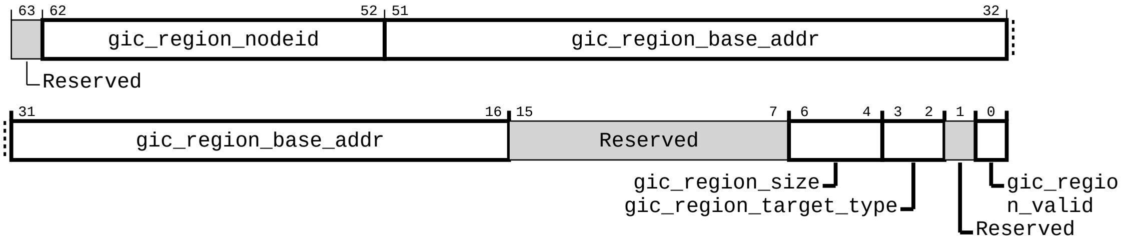


Table 5-185: gic_mem_region_reg attributes

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: Reserved 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

5.3.7.18 sys_cache_grp_cal_mode_reg

Configures the HN-F CAL mode support for all System Cache Groups (SCGs).

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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-170: sys_cache_grp_cal_mode_reg

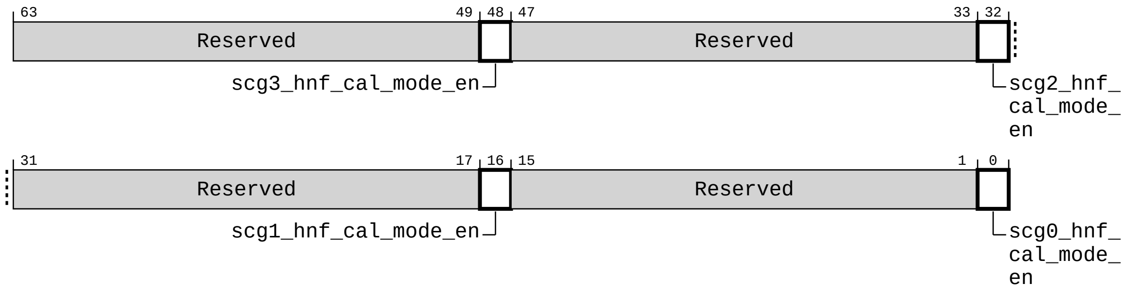


Table 5-186: sys_cache_grp_cal_mode_reg attributes

Bits	Name	Description	Type	Reset
[63:49]	Reserved	Reserved	RO	-
[48]	scg3_hnf_cal_mode_en	Enables support for HN-F CAL for SCG 3	RW	1'b0
[47:33]	Reserved	Reserved	RO	-
[32]	scg2_hnf_cal_mode_en	Enables support for HN-F CAL for SCG 2	RW	1'b0
[31:17]	Reserved	Reserved	RO	-
[16]	scg1_hnf_cal_mode_en	Enables support for HN-F CAL for SCG 1	RW	1'b0
[15:1]	Reserved	Reserved	RO	-
[0]	scg0_hnf_cal_mode_en	Enables support for HN-F CAL for SCG 0	RW	1'b0

5.3.7.19 sys_cache_grp_sn_sam_cfg0-1

There are 2 iterations of this register, parameterized by the index from 0 to 1. Configures top address bits for SN SAM System Cache Groups (SCGs) #{index*2 + 0} 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

16'h1140 + (8 × #[0, 1])

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

Figure 5-171: sys_cache_grp_sn_sam_cfg0-1

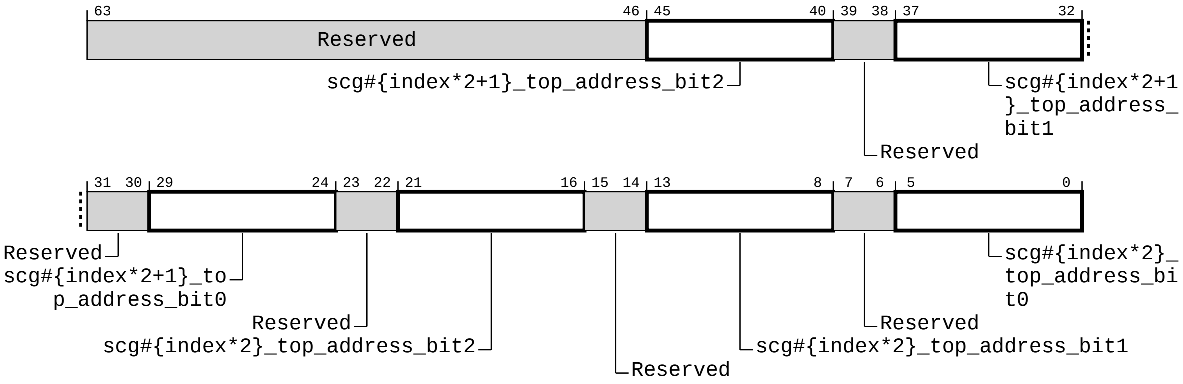

63
46
45
40
39
38
37
32
Reserved
scg#{index*2+1}_top_address_bit2
scg#{index*2+1}_top_address_bit1
Reserved
31
30
29
24
23
22
21
16
15
14
13
8
7
6
5
0
Reserved
scg#{index*2+1}_top_address_bit0
Reserved
scg#{index*2}_top_address_bit2
Reserved
scg#{index*2}_top_address_bit1
Reserved
scg#{index*2}_top_address_bit0

Table 5-187: sys_cache_grp_sn_sam_cfg0-1 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 SCG #{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 SCG #{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 SCG #{index*2+1}	RW	6'h00
[23:22]	Reserved	Reserved	RO	-
[21:16]	scg#{index*2}_top_address_bit2	Top address bit 2 for SCG #{index*2}	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	scg#{index*2}_top_address_bit1	Top address bit 1 for SCG #{index*2}	RW	6'h00
[7:6]	Reserved	Reserved	RO	-
[5:0]	scg#{index*2}_top_address_bit0	Top address bit 0 for SCG #{index*2}	RW	6'h00

5.3.7.20 sys_cache_grp_hn_cpa_en_reg

Configures CCIX Port Aggregation (CPA) mode for hashed HN 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

Figure 5-172: sys_cache_grp_hn_cpa_en_reg

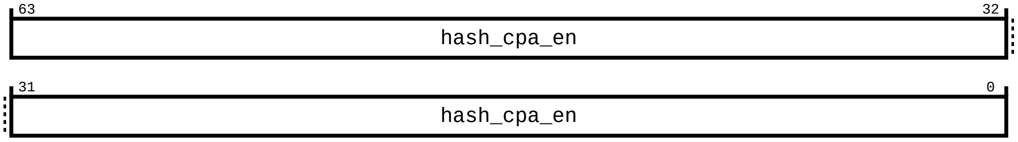


Table 5-188: sys_cache_grp_hn_cpa_en_reg attributes

Bits	Name	Description	Type	Reset
[63:0]	hash_cpa_en	Enable CPA for each hashed node ID	RW	64'h0000000000000000

5.3.7.21 sys_cache_grp_hn_cpa_grp_reg

Configures CCIX Port Aggregation Group (CPAG) ID for each System Cache Group (SCG).

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

Figure 5-173: sys_cache_grp_hn_cpa_grp_reg

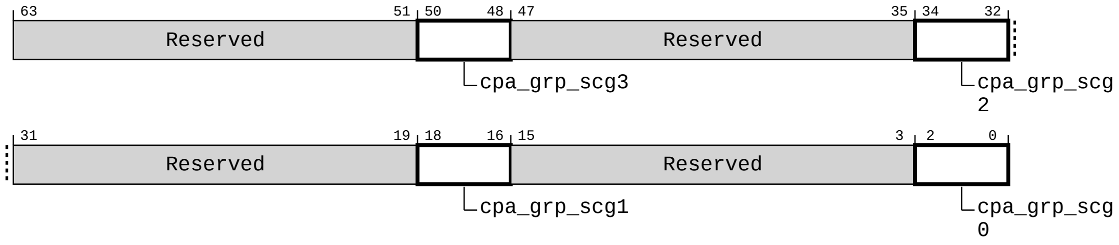


Table 5-189: sys_cache_grp_hn_cpa_grp_reg attributes

Bits	Name	Description	Type	Reset
[63:51]	Reserved	Reserved	RO	-
[50:48]	cpa_grp_scg3	Specifies CPAG ID for SCG 3: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'b000
[47:35]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[34:32]	cpa_grp_scg2	Specifies CPAG ID for SCG 2: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'b000
[31:19]	Reserved	Reserved	RO	-
[18:16]	cpa_grp_scg1	Specifies CPAG ID for SCG 1: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'b000
[15:3]	Reserved	Reserved	RO	-
[2:0]	cpa_grp_scg0	Specifies CPAG ID for SCG 0: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'b000

5.3.7.22 cml_port_aggr_mode_ctrl_reg

Configures the CCIX Port Aggregation (CPA) 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

Figure 5-174: cml_port_aggr_mode_ctrl_reg

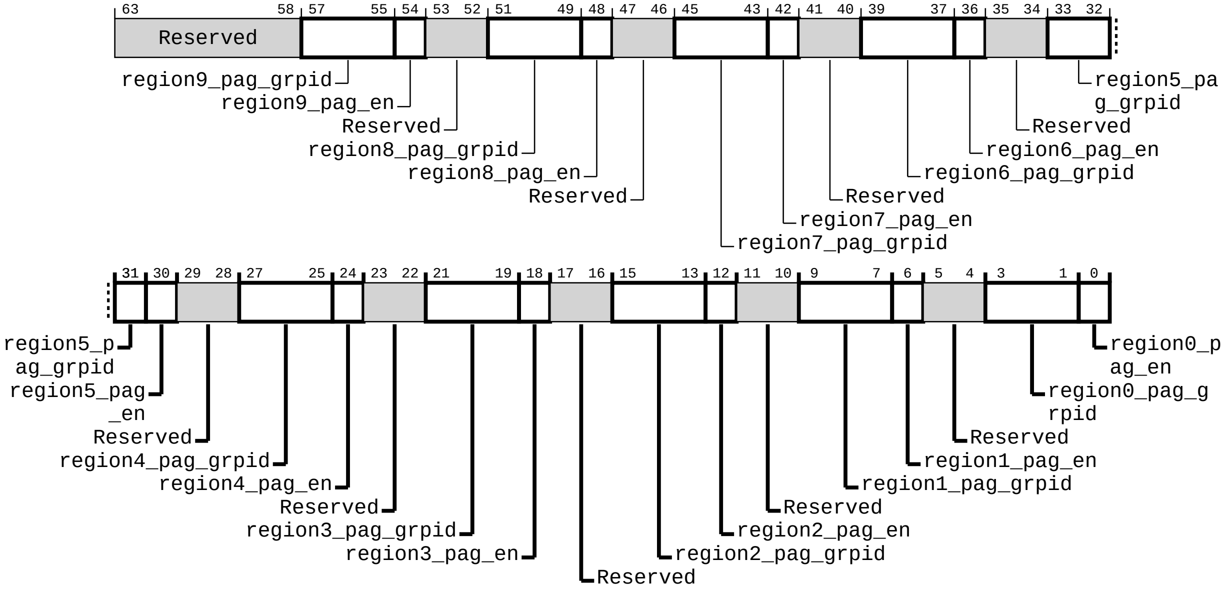


Table 5-190: cml_port_aggr_mode_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:55]	region9_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[54]	region9_pag_en	Enables the CPA mode for non-hashed memory region 9	RW	1'b0
[53:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:49]	region8_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[48]	region8_pag_en	Enables the CPA mode for non-hashed memory region 8	RW	1'b0
[47:46]	Reserved	Reserved	RO	-
[45:43]	region7_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[42]	region7_pag_en	Enables the CPA mode for non-hashed memory region 7	RW	1'b0
[41:40]	Reserved	Reserved	RO	-
[39:37]	region6_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[36]	region6_pag_en	Enables the CPA mode for non-hashed memory region 6	RW	1'b0
[35:34]	Reserved	Reserved	RO	-
[33:31]	region5_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[30]	region5_pag_en	Enables the CPA mode for non-hashed memory region 5	RW	1'b0

Bits	Name	Description	Type	Reset
[29:28]	Reserved	Reserved	RO	-
[27:25]	region4_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[24]	region4_pag_en	Enables the CPA mode for non-hashed memory region 4	RW	1'b0
[23:22]	Reserved	Reserved	RO	-
[21:19]	region3_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[18]	region3_pag_en	Enables the CPA mode for non-hashed memory region 3	RW	1'b0
[17:16]	Reserved	Reserved	RO	-
[15:13]	region2_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[12]	region2_pag_en	Enables the CPA mode for non-hashed memory region 2	RW	1'b0
[11:10]	Reserved	Reserved	RO	-
[9:7]	region1_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0

Bits	Name	Description	Type	Reset
[6]	region1_pag_en	Enables the CPA mode for non-hashed memory region 1	RW	1'b0
[5:4]	Reserved	Reserved	RO	-
[3:1]	region0_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[0]	region0_pag_en	Enables the CPA mode for non-hashed memory region 0	RW	1'b0

5.3.7.23 cml_port_aggr_mode_ctrl_reg1

Configures the CCIX Port Aggregation (CPA) modes for non-hashed memory regions 8 through 19.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h11A8

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

Figure 5-175: cml_port_aggr_mode_ctrl_reg1

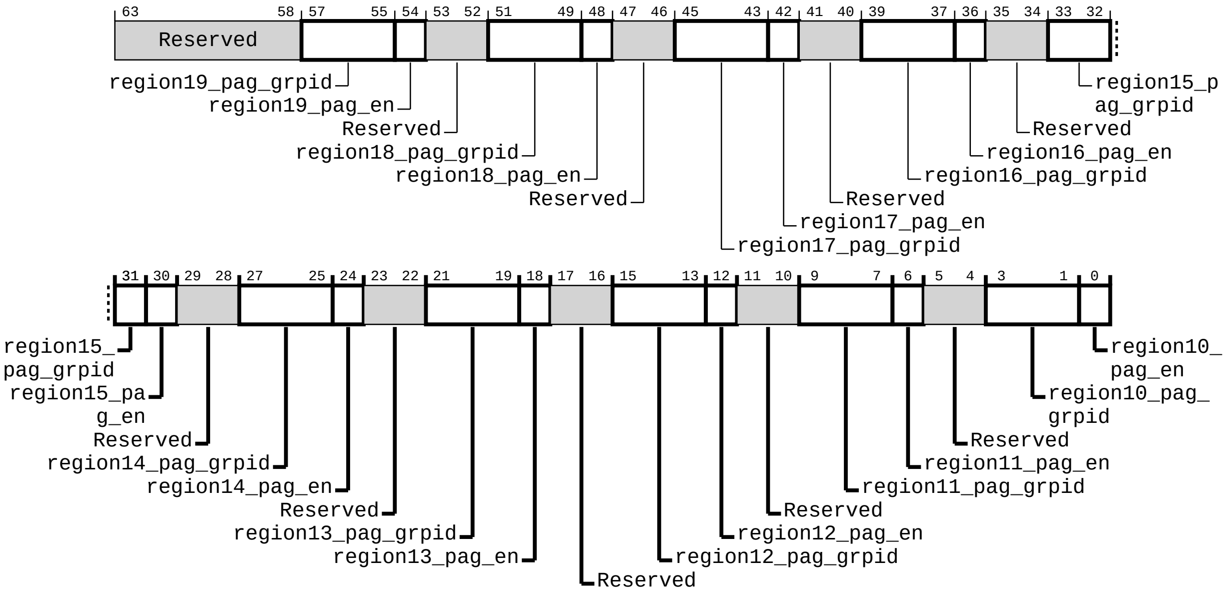


Table 5-191: cml_port_aggr_mode_ctrl_reg1 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:55]	region19_pag_grpid	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[54]	region19_pag_en	Enables the CPA mode for non-hashed memory region 19	RW	1'b0
[53:52]	Reserved	Reserved	RO	-
[51:49]	region18_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0

Bits	Name	Description	Type	Reset
[48]	region18_pag_en	Enables the CPA mode for non-hashed memory region 18	RW	1'b0
[47:46]	Reserved	Reserved	RO	-
[45:43]	region17_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[42]	region17_pag_en	Enables the CPA mode for non-hashed memory region 17	RW	1'b0
[41:40]	Reserved	Reserved	RO	-
[39:37]	region16_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[36]	region16_pag_en	Enables the CPA mode for non-hashed memory region 16	RW	1'b0
[35:34]	Reserved	Reserved	RO	-
[33:31]	region15_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[30]	region15_pag_en	Enables the CPA mode for non-hashed memory region 15	RW	1'b0
[29:28]	Reserved	Reserved	RO	-
[27:25]	region14_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[24]	region14_pag_en	Enables the CPA mode for non-hashed memory region 14	RW	1'b0
[23:22]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[21:19]	region13_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[18]	region13_pag_en	Enables the CPA mode for non-hashed memory region 13	RW	1'b0
[17:16]	Reserved	Reserved	RO	-
[15:13]	region12_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[12]	region12_pag_en	Enables the CPA mode for non-hashed memory region 12	RW	1'b0
[11:10]	Reserved	Reserved	RO	-
[9:7]	region11_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[6]	region11_pag_en	Enables the CPA mode for non-hashed memory region 11	RW	1'b0
[5:4]	Reserved	Reserved	RO	-
[3:1]	region10_pag_grpid	Specifies CPAG ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0
[0]	region10_pag_en	Enables the CPA mode for non-hashed memory region 10	RW	1'b0

5.3.7.24 cml_port_aggr_grp0-4_add_mask

There are 5 iterations of this register, parameterized by the index from 0 to 4. Configures the CCIX Port Aggregation Group (CPAG) address mask for CPAG #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h11C0 + (8 × #[0, 1, ... 4])

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

Figure 5-176: cml_port_aggr_grp0-4_add_mask

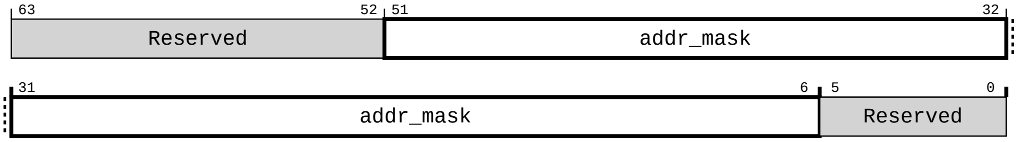


Table 5-192: cml_port_aggr_grp0-4_add_mask attributes

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	-

5.3.7.25 cml_port_aggr_grp_reg0-1

There are 2 iterations of this register, parameterized by the index from 0 to 1. Configures the CCIX Port Aggregation Group (CPAG) port Node IDs.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h11F0 + (8 × #[0, 1])

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

Figure 5-177: cml_port_aggr_grp_reg0-1

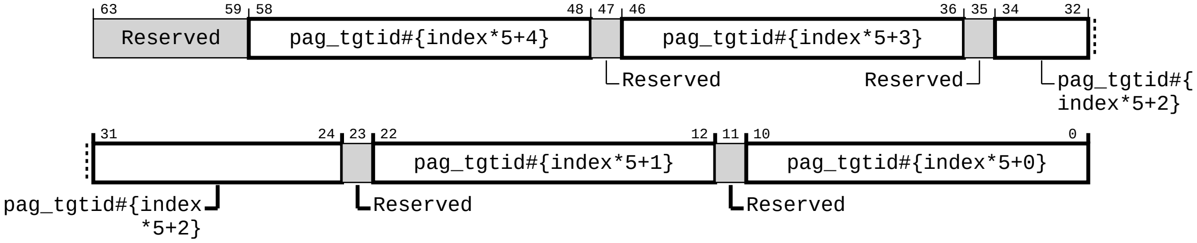


Table 5-193: cml_port_aggr_grp_reg0-1 attributes

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

Bits	Name	Description	Type	Reset
[11]	Reserved	Reserved	RO	-
[10:0]	pag_tgtid#{index*5+0}	Specifies target ID #{index*5+0} for CPAG	RW	11'b0

5.3.7.26 cml_port_aggr_ctrl_reg

Configures the CCIX Port Aggregation Group (CPAG) 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

Figure 5-178: cml_port_aggr_ctrl_reg

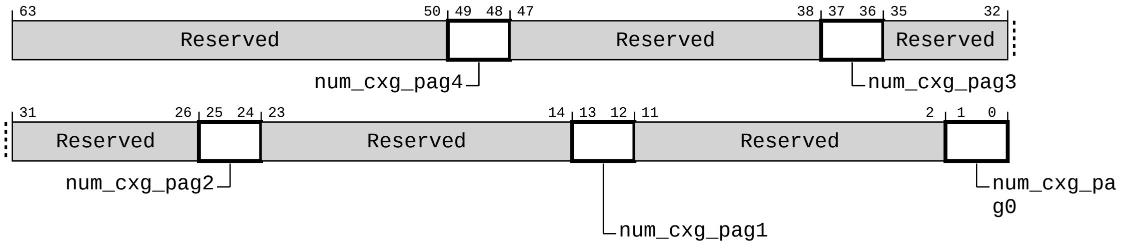


Table 5-194: cml_port_aggr_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[49:48]	num_cxg_pag4	Specifies the number of CXRAs in CPAG 4 Constraint: Can use pag_tgtid8 through pag_tgtid9 of cml_port_aggr_grp_reg1. 2'b00: One port used 2'b01: Two ports used 2'b10: Reserved 2'b11: Reserved	RW	2'b0
[47:38]	Reserved	Reserved	RO	-
[37:36]	num_cxg_pag3	Specifies the number of CXRAs in CPAG 3 Constraint: Can use pag_tgtid6 through pag_tgtid7 of cml_port_aggr_grp_reg1. 2'b00: One port used 2'b01: Two ports used 2'b10: Reserved 2'b11: Reserved	RW	2'b0
[35:26]	Reserved	Reserved	RO	-
[25:24]	num_cxg_pag2	Specifies the number of CXRAs in CPAG 2 Constraint: Can use pag_tgtid4 through pag_tgtid7 of cml_port_aggr_grp_reg[0,1]. 2'b00: One port used 2'b01: Two ports used 2'b10: Four ports used 2'b11: Reserved	RW	2'b0
[23:14]	Reserved	Reserved	RO	-
[13:12]	num_cxg_pag1	Specifies the number of CXRAs in CPAG 1 Constraint: Can use pag_tgtid2 through pag_tgtid3 of cml_port_aggr_grp_reg0. 2'b00: One port used 2'b01: Two ports used 2'b10: Reserved 2'b11: Reserved	RW	2'b0
[11:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1:0]	num_cxg_pag0	Specifies the number of CXRAs in CPAG 0 Constraint: Can use pag_tgtid0 through pag_tgtid7 of cml_port_aggr_grp_reg[0,1]. 2'b00: One port used 2'b01: Two ports used 2'b10: Four ports used 2'b11: Eight ports used	RW	2'b0

5.3.7.27 sam_qos_mem_region_reg0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Configures the QoS value for memory region #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1280 + (8 × #[0, 1, ... 7])

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

Figure 5-179: sam_qos_mem_region_reg0-7

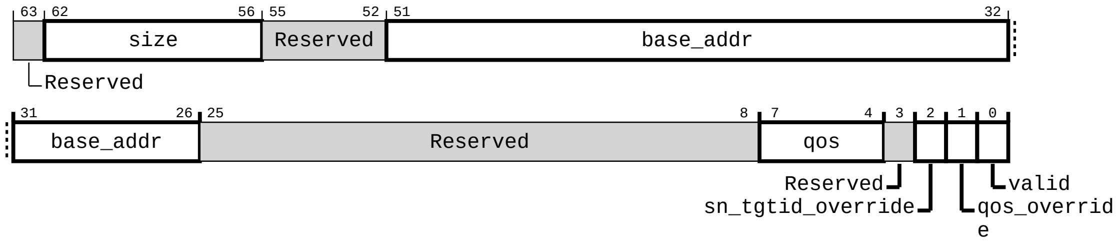


Table 5-195: sam_qos_mem_region_reg0-7 attributes

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	-
[51:26]	base_addr	Bits[51:26] of base address of the memory region. CONSTRAINT: Must be an integer multiple of region size.	RW	26'b00000000000000000000000000
[25:8]	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 target ID 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
[0]	valid	QoS memory region valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'b0

5.3.7.28 sys_cache_grp_region0_sn_nodeid_reg0-15

There are 16 iterations of this register, parameterized by the index from 0 to 15. Configures hashed node IDs for memory region 0 of the System Cache Group (SCG). 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'h1400 + (8 × #[0, 1, ... 15])

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

Figure 5-180: sys_cache_grp_region0_sn_nodeid_reg0-15

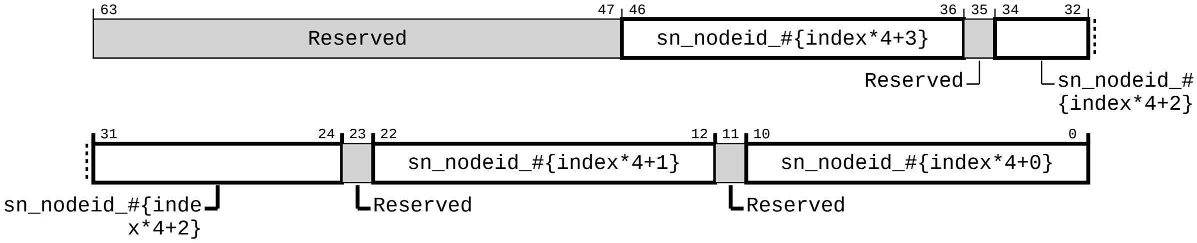


Table 5-196: sys_cache_grp_region0_sn_nodeid_reg0-15 attributes

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	-

Bits	Name	Description	Type	Reset
[10:0]	sn_nodeid_#{index*4+0}	Hashed target SN node ID #{index*4+0}	RW	11'b00000000000

5.3.7.29 sys_cache_grp_region1_sn_nodeid_reg0-15

There are 16 iterations of this register, parameterized by the index from 0 to 15. Configures hashed node IDs for memory region 1 of the System Cache Group (SCG). 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'h1500 + (8 × #[0, 1, ... 15])

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

Figure 5-181: sys_cache_grp_region1_sn_nodeid_reg0-15

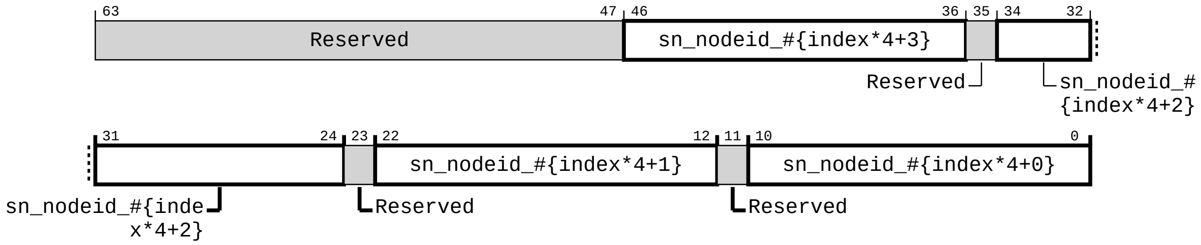


Table 5-197: sys_cache_grp_region1_sn_nodeid_reg0-15 attributes

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	-

Bits	Name	Description	Type	Reset
[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+0}	Hashed target SN node ID #{index*4+0}	RW	11'b00000000000

5.3.8 HN-F MPAM_S register descriptions

This section lists the HN-F MPAM_S registers.

5.3.8.1 por_hnf_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

Figure 5-182: por_hnf_mpam_s_node_info

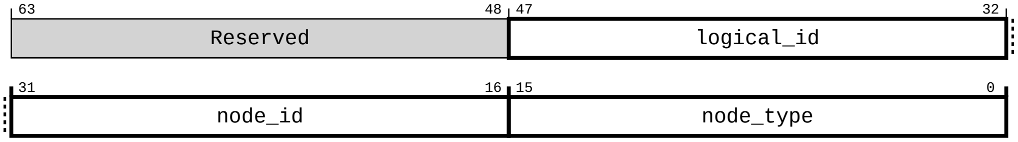


Table 5-198: por_hnf_mpam_s_node_info attributes

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-650 node type identifier	RO	16'h0008

5.3.8.2 por_hnf_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

Figure 5-183: por_hnf_mpam_s_child_info

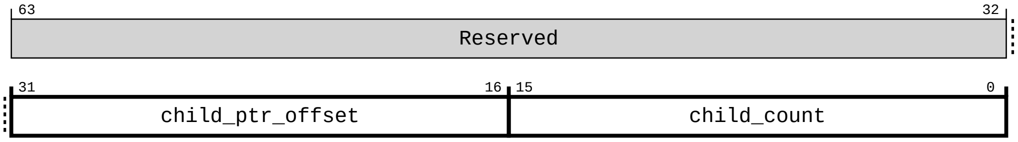


Table 5-199: por_hnf_mpam_s_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.8.3 por_hnf_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. Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

Figure 5-184: por_hnf_mpam_s_secure_register_groups_override

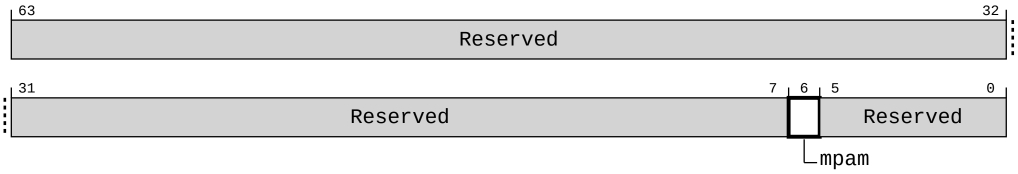


Table 5-200: por_hnf_mpam_s_secure_register_groups_override attributes

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	-

5.3.8.4 por_hnf_mpam_sidr

MPAM features Secure ID register. This is a 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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-185: por_hnf_mpam_sidr

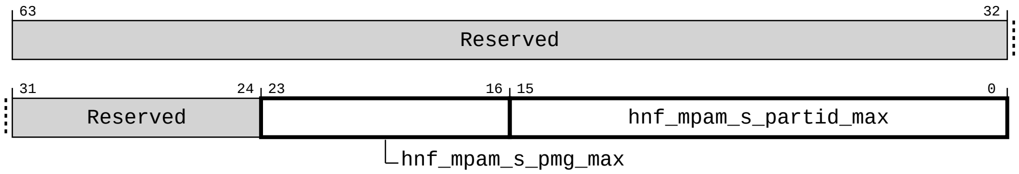


Table 5-201: por_hnf_mpam_sidr attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hnf_mpam_s_pmg_max	Maximum value of Secure PMG that is supported by this HN-F	RO	Configuration dependent
[15:0]	hnf_mpam_s_partid_max	Maximum value of Secure PARTID that is supported by this HN-F	RO	Configuration dependent

5.3.8.5 por_hnf_s_mpam_ecr

MPAM Error Control Register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-186: por_hnf_s_mpam_ecr

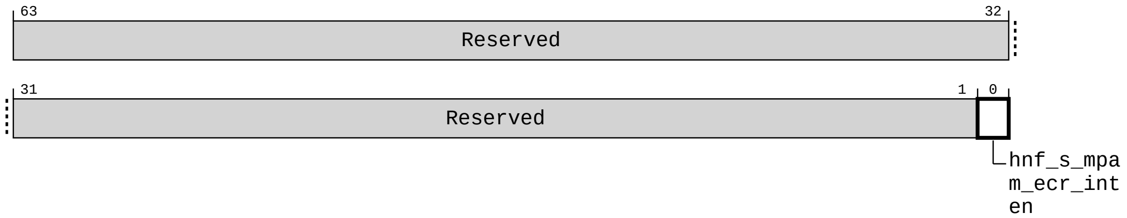


Table 5-202: por_hnf_s_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hnf_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

5.3.8.6 por_hnf_s_mpam_esr

MPAM Error Status Register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-187: por_hnf_s_mpam_esr

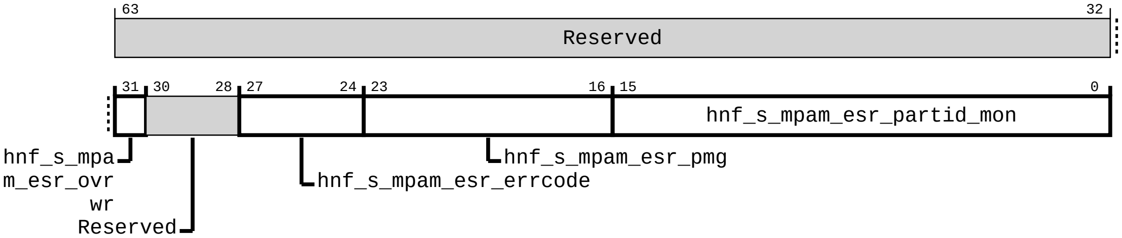


Table 5-203: por_hnf_s_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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. It is only reached when software writes this combination into this register.	RW	1'h0
[30:28]	Reserved	Reserved	RO	-
[27:24]	hnf_s_mpam_esr_errcode	Error code	RW	4'h0
[23:16]	hnf_s_mpam_esr_pmg	PMG is captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hnf_s_mpam_esr_partid_mon	PARTID is captured if the error code captures PARTID. MON selector captured if the error code captures MON. Otherwise 0x0000.	RW	16'h0

5.3.8.7 por_hnf_s_mpamcfg_part_sel

MPAM partition configuration selection register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-188: por_hnf_s_mpamcfg_part_sel

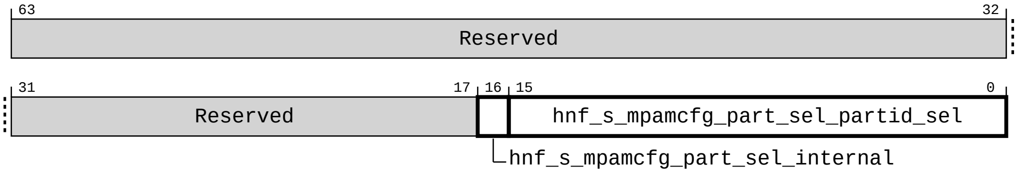

63
32
Reserved
31
17
16
15
0
Reserved
hnf_s_mpamcfg_part_sel_partid_sel
hnf_s_mpamcfg_part_sel_internal

Table 5-204: por_hnf_s_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hnf_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 interpret PARTID_SEL	RW	1'h0
[15:0]	hnf_s_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure	RW	16'h0

5.3.8.8 por_hnf_s_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-189: por_hnf_s_mpamcfg_cmax

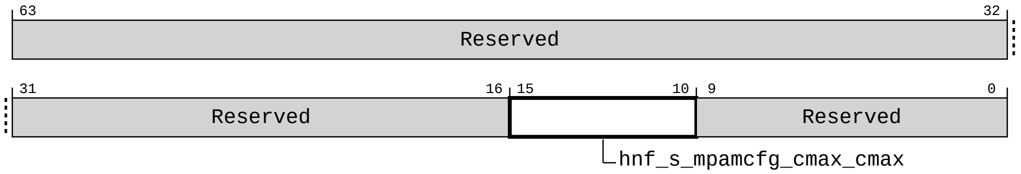


Table 5-205: por_hnf_s_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:10]	hnf_s_mpamcfg_cmax_cmax	Maximum cache capacity used in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL	RW	6'h3f
[9:0]	Reserved	Reserved	RO	-

5.3.8.9 por_hnf_s_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-190: por_hnf_s_mpamcfg_mbw_min

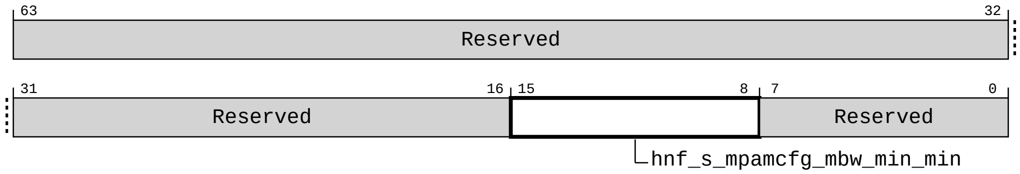


Table 5-206: por_hnf_s_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	hnf_s_mpamcfg_mbw_min_min	Memory bandwidth minimum that is allocated to the partition that is selected by MPAMCFG_PART_SEL	RW	8'h0
[7:0]	Reserved	Reserved	RO	-

5.3.8.10 por_hnf_s_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-191: por_hnf_s_mpamcfg_mbw_max

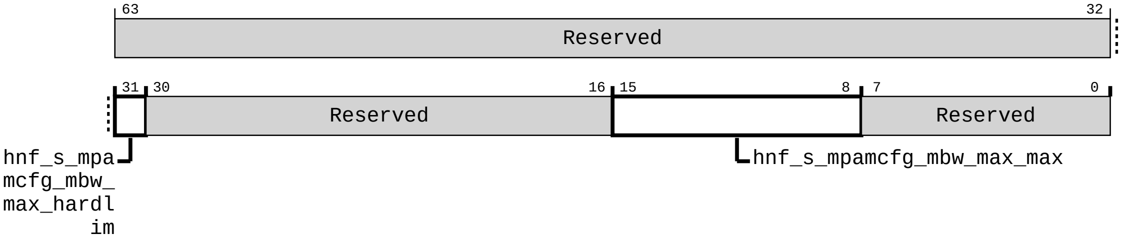


Table 5-207: por_hnf_s_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_s_mpamcfg_mbw_max_hardlim	0: When the bandwidth maximum is exceeded, the partition can contend with a low preference for downstream bandwidth beyond its maximum bandwidth. 1: When the bandwidth maximum is exceeded, the partition cannot use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0

Bits	Name	Description	Type	Reset
[30:16]	Reserved	Reserved	RO	-
[15:8]	hnf_s_mpamcfg_mbw_max_max	Memory bandwidth maximum that is allocated to the partition that is selected by MPAMCFG_PART_SEL.	RW	8'h0
[7:0]	Reserved	Reserved	RO	-

5.3.8.11 por_hnf_s_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-192: por_hnf_s_mpamcfg_mbw_winwd

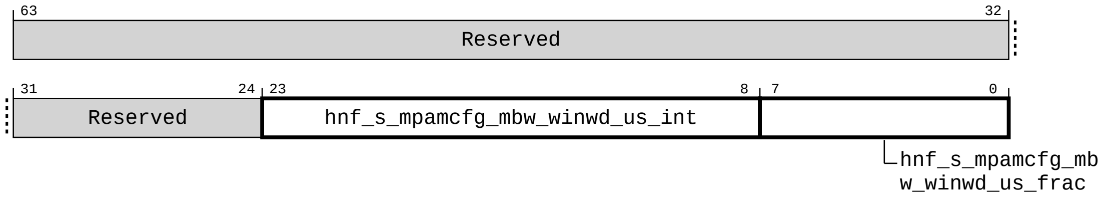

63
32
Reserved
31
24
23
8
7
0
Reserved
hnf_s_mpamcfg_mbw_winwd_us_int
hnf_s_mpamcfg_mbw_winwd_us_frac

Table 5-208: por_hnf_s_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[23:8]	hnf_s_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period, integer microseconds.	RW	16'h0
[7:0]	hnf_s_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period, fractions of a microsecond.	RW	8'h0

5.3.8.12 por_hnf_s_mpamcfg_pri

MPAM priority partitioning configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-193: por_hnf_s_mpamcfg_pri

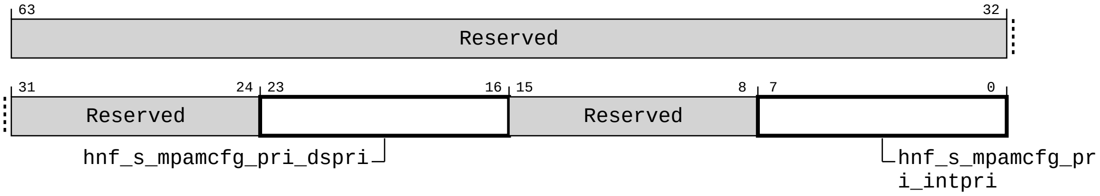


Table 5-209: por_hnf_s_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hnf_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 that is selected by MPAMCFG_PART_SEL.	RW	8'h0

Bits	Name	Description	Type	Reset
[15:8]	Reserved	Reserved	RO	-
[7:0]	hnf_s_mpamcfg_pri_intpri	If HAS_INTPRI is 1, this field is a priority value that is applied internally inside this memory system component for transactions of the partition that is selected by MPAMCFG_PART_SEL.	RW	8'h0

5.3.8.13 por_hnf_s_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-194: por_hnf_s_mpamcfg_mbw_prop

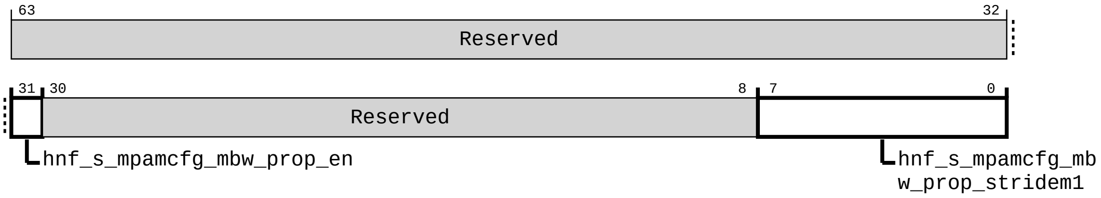


Table 5-210: por_hnf_s_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hnf_s_mpamcfg_mbw_prop_en	0: The selected partition is not regulated by proportional stride bandwidth partitioning. 1: The selected partition has its bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.	RW	1'h0
[30:8]	Reserved	Reserved	RO	-
[7:0]	hnf_s_mpamcfg_mbw_prop_stridem1	Normalized cost of bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	8'h0

5.3.8.14 por_hnf_s_mpamcfg_intpartid

MPAM internal partition narrowing configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-195: por_hnf_s_mpamcfg_intpartid

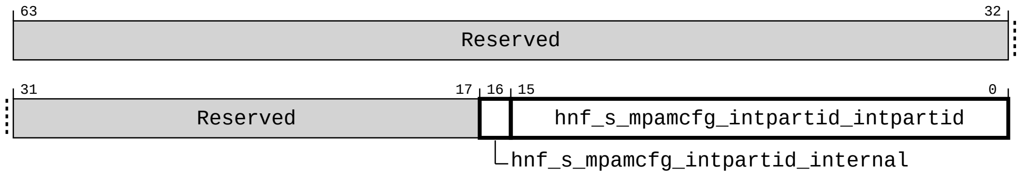

63
32
Reserved
31
17
16
15
0
Reserved
hnf_s_mpamcfg_intpartid_intpartid
hnf_s_mpamcfg_intpartid_internal

Table 5-211: por_hnf_s_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hnf_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]	hnf_s_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

5.3.8.15 por_hnf_s_msmon_cfg_mon_sel

Memory system performance monitor selection register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-196: por_hnf_s_msmon_cfg_mon_sel

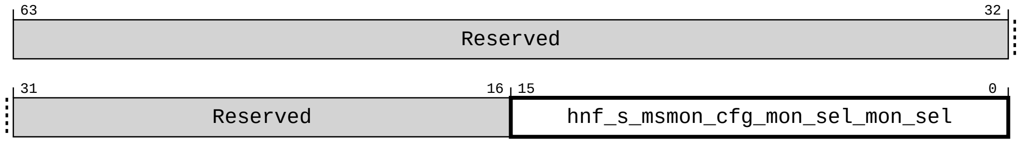


Table 5-212: por_hnf_s_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_s_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure	RW	16'h0

5.3.8.16 por_hnf_s_msmon_capt_evnt

Memory system performance monitoring capture event generation register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-197: por_hnf_s_msmon_capt_evnt

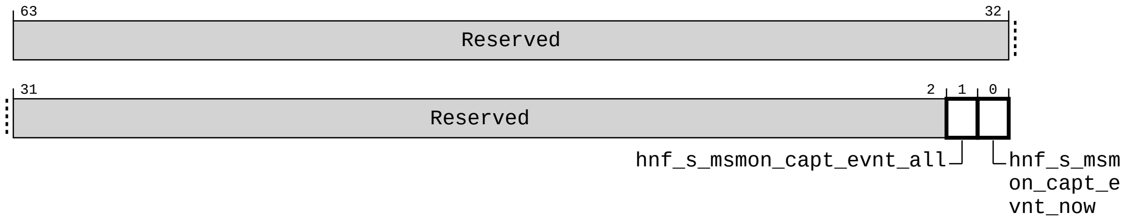


Table 5-213: por_hnf_s_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1]	hnf_s_msmon_capt_evnt_all	In the Secure version, if ALL is written as 1 and NOW is also written as 1, it signals 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, it signals 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, it signals a capture event to Non-secure monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hnf_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

5.3.8.17 por_hnf_s_msmon_cfg_csu_flt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-198: por_hnf_s_msmon_cfg_csu_flt

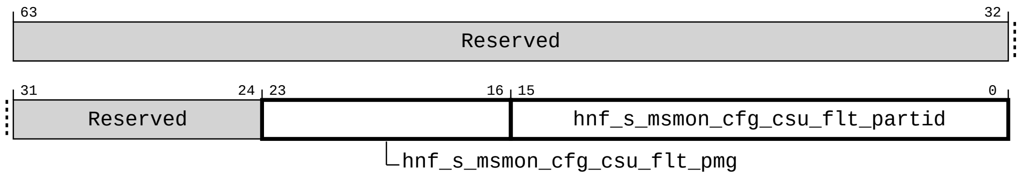


Table 5-214: por_hnf_s_msmon_cfg_csu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hnf_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 that are labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hnf_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 that are labelled with both the configured PARTID and PMG.	RW	16'h0

5.3.8.18 por_hnf_s_msmon_cfg_csu_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-199: por_hnf_s_msmon_cfg_csu_ctl

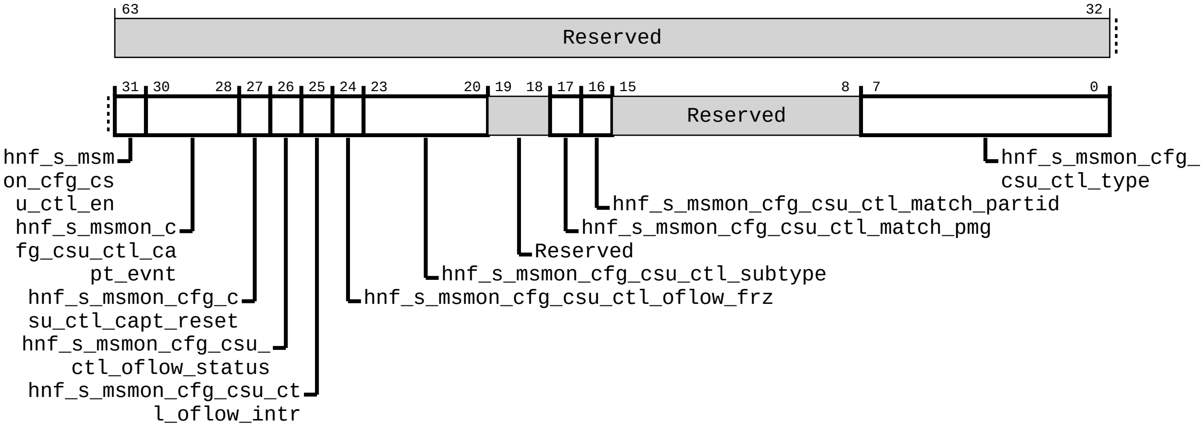

63
32
Reserved
31
30
28
27
26
25
24
23
20
19
18
17
16
15
8
7
0
Reserved
hnf_s_msmon_cfg_csu_ctl_en
hnf_s_msmon_cfg_csu_ctl_capt_evnt
hnf_s_msmon_cfg_csu_ctl_capt_reset
hnf_s_msmon_cfg_csu_ctl_oflow_status
hnf_s_msmon_cfg_csu_ctl_oflow_intr
hnf_s_msmon_cfg_csu_ctl_oflow_frz
hnf_s_msmon_cfg_csu_ctl_subtype
Reserved
hnf_s_msmon_cfg_csu_ctl_match_pmg
hnf_s_msmon_cfg_csu_ctl_match_partid
hnf_s_msmon_cfg_csu_ctl_type

Table 5-215: por_hnf_s_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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]	hnf_s_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	hnf_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]	hnf_s_msmon_cfg_csu_ctl_oflow_intr	0: No interrupt 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hnf_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]	hnf_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]	hnf_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

Bits	Name	Description	Type	Reset
[16]	hnf_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]	hnf_s_msmon_cfg_csu_ctl_type	Read-only: Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	8'h43

5.3.8.19 por_hnf_s_msmon_cfg_mbwu_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-200: por_hnf_s_msmon_cfg_mbwu_flt

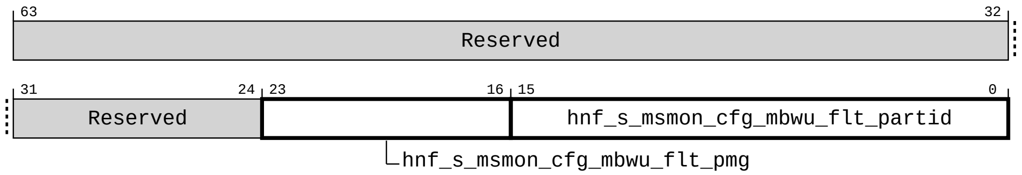


Table 5-216: por_hnf_s_msmon_cfg_mbwu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hnf_s_msmon_cfg_mbwu_flt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor that is selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests that are labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hnf_s_msmon_cfg_mbwu_flt_partid	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor that is selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests that are labelled with both the configured PARTID and PMG.	RW	16'h0

5.3.8.20 por_hnf_s_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-201: por_hnf_s_msmon_cfg_mbwu_ctl

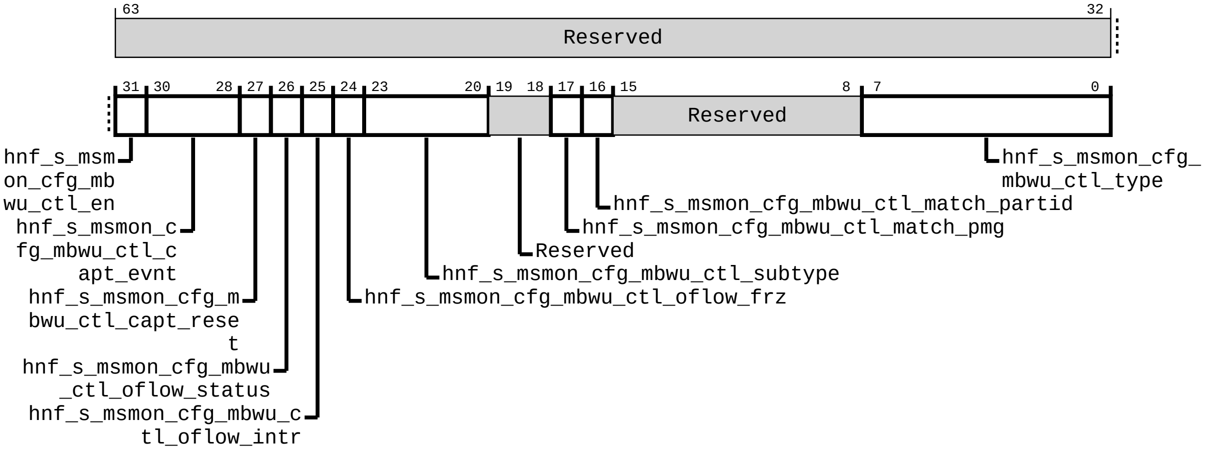

63
32
Reserved
31
30
28
27
26
25
24
23
20
19
18
17
16
15
8
7
0
Reserved
hnf_s_msmon_cfg_mbwu_ctl_en
hnf_s_msmon_cfg_mbwu_ctl_capt_evnt
hnf_s_msmon_cfg_mbwu_ctl_capt_reset
hnf_s_msmon_cfg_mbwu_ctl_oflow_status
hnf_s_msmon_cfg_mbwu_ctl_oflow_intr
hnf_s_msmon_cfg_mbwu_ctl_oflow_frz
hnf_s_msmon_cfg_mbwu_ctl_subtype
Reserved
hnf_s_msmon_cfg_mbwu_ctl_match_pmg
hnf_s_msmon_cfg_mbwu_ctl_match_partid
hnf_s_msmon_cfg_mbwu_ctl_type

Table 5-217: por_hnf_s_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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]	hnf_s_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hnf_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]	hnf_s_msmon_cfg_mbwu_ctl_oflow_intr	0: No interrupt 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hnf_s_msmon_cfg_mbwu_ctl_oflow_frz	0: Monitor count wraps on overflow. 1: Monitor count freezes on overflow. The frozen value can be 0 or another value if the monitor has overflowed with an increment larger than 1.	RW	1'h0

Bits	Name	Description	Type	Reset
[23:20]	hnf_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 behavior of a monitor with SUBTYPE set to a reserved value is UNPREDICTABLE.	RW	4'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hnf_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]	hnf_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]	hnf_s_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

5.3.8.21 por_hnf_s_msmon_csu

Memory system performance monitor cache storage usage monitor register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-202: por_hnf_s_msmon_csu

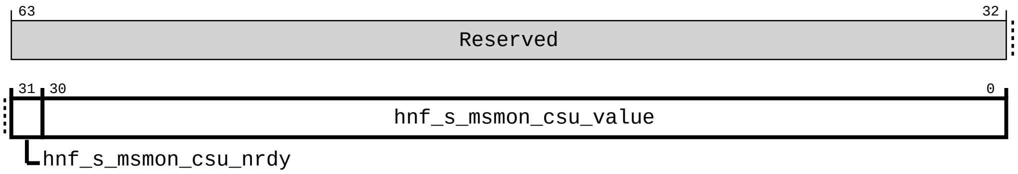


Table 5-218: por_hnf_s_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.8.22 por_hnf_s_msmon_csu_capture

Memory system performance monitor cache storage usage capture register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-203: por_hnf_s_msmon_csu_capture

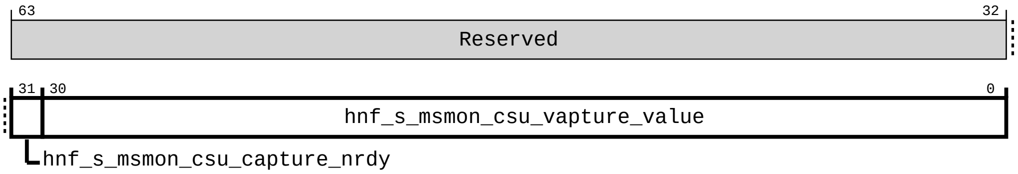


Table 5-219: por_hnf_s_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.8.23 por_hnf_s_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-204: por_hnf_s_msmon_mbwu

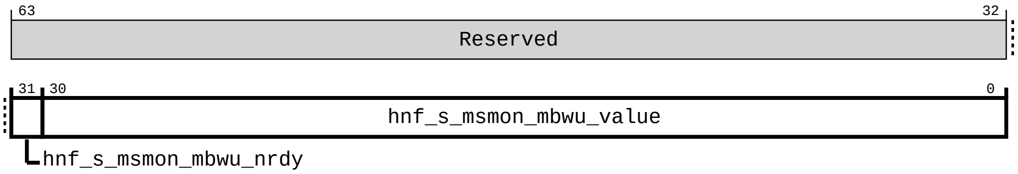


Table 5-220: por_hnf_s_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.8.24 por_hnf_s_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-205: por_hnf_s_msmon_mbwu_capture

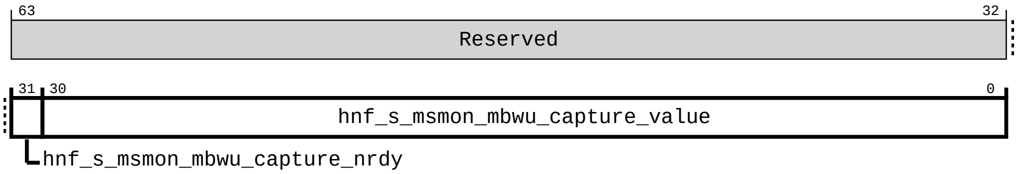


Table 5-221: por_hnf_s_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.8.25 por_hnf_s_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked seperately for S and 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

por_hnf_mpam_s_secure_register_groups_override.mpam

Usage constraints

Only accessible by Secure accesses. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-206: por_hnf_s_mpamcfg_cpbm

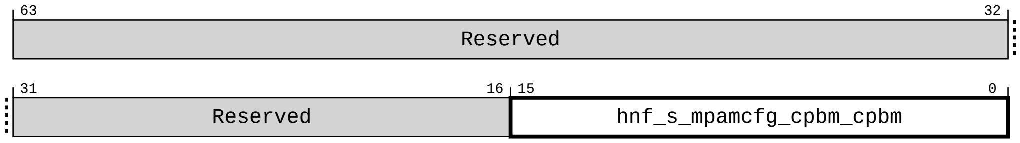


Table 5-222: por_hnf_s_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_s_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition that is selected by MPAMCFG_PART_SEL. Note: CPBM cannot be all zeros for any PARTID.	RW	16'hFFFF

5.3.9 Configuration master register descriptions

This section lists the configuration registers.

5.3.9.1 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

Figure 5-207: por_cfgm_node_info

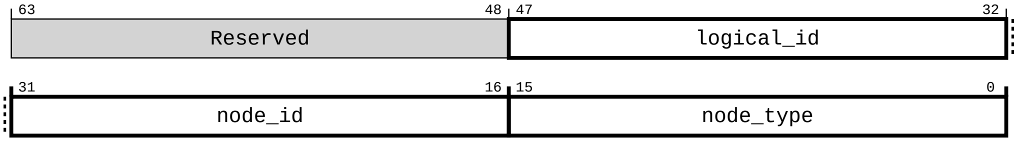


Table 5-223: por_cfgm_node_info attributes

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-650 node type identifier	RO	16'h0002

5.3.9.2 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

Figure 5-208: por_cfgm_periph_id_0_periph_id_1

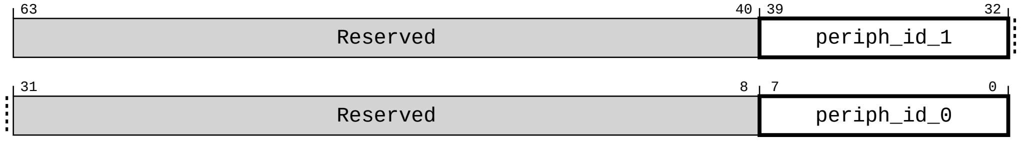


Table 5-224: por_cfgm_periph_id_0_periph_id_1 attributes

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

5.3.9.3 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

Figure 5-209: por_cfgm_periph_id_2_periph_id_3

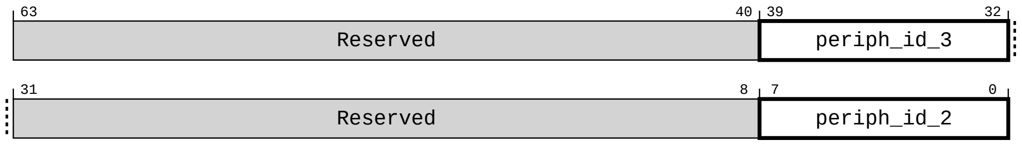


Table 5-225: por_cfgm_periph_id_2_periph_id_3 attributes

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 r1p1 0x3 r2p0 [3] JEDEC JEP106 identity code, 1'b1 [2:0] JEP106 identity code [6:4], 0'b011	RO	Configuration dependent

5.3.9.4 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

Figure 5-210: por_cfgm_periph_id_4_periph_id_5

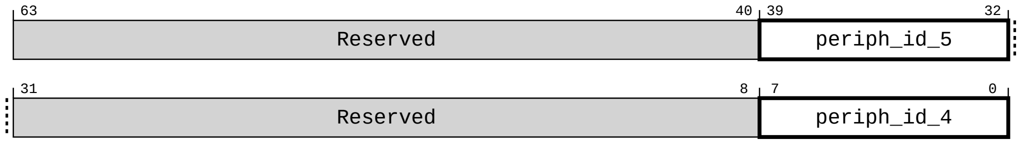


Table 5-226: por_cfgm_periph_id_4_periph_id_5 attributes

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

5.3.9.5 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

Figure 5-211: por_cfgm_periph_id_6_periph_id_7

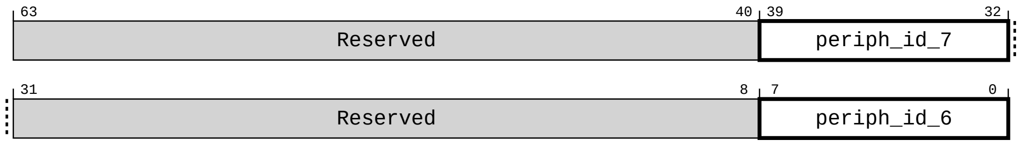


Table 5-227: por_cfgm_periph_id_6_periph_id_7 attributes

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

5.3.9.6 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

Figure 5-212: por_cfgm_component_id_0_component_id_1

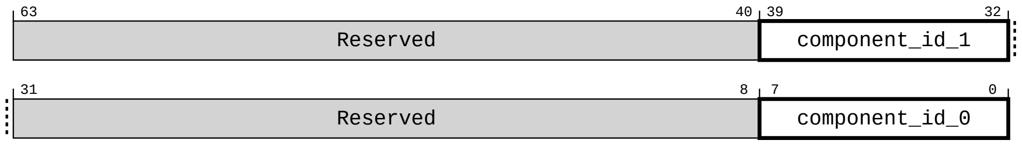


Table 5-228: por_cfgm_component_id_0_component_id_1 attributes

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

5.3.9.7 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

Figure 5-213: por_cfgm_component_id_2_component_id_3

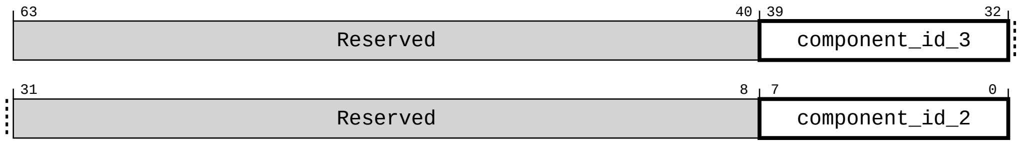


Table 5-229: por_cfgm_component_id_2_component_id_3 attributes

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

5.3.9.8 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

Figure 5-214: por_cfgm_child_info

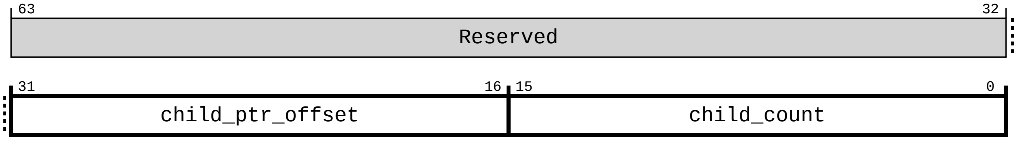


Table 5-230: por_cfgm_child_info attributes

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

5.3.9.9 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

Figure 5-215: por_cfgm_secure_access

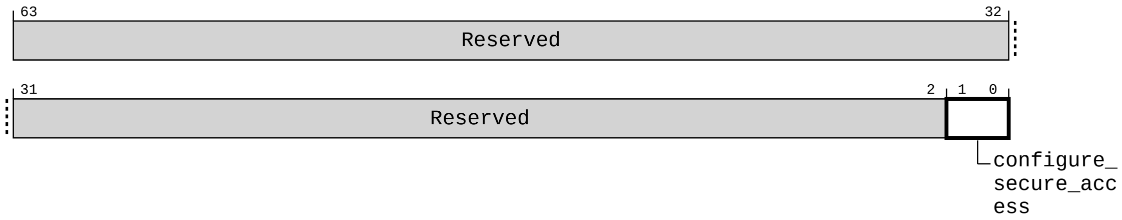


Table 5-231: por_cfgm_secure_access attributes

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

5.3.9.10 por_cfgm_errgsr_mxp_0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the XP <n> Secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3000 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-216: por_cfgm_errgsr_mxp_0-7

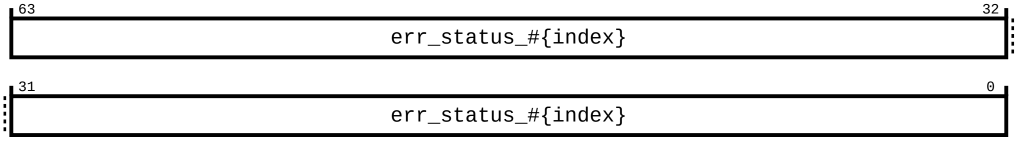


Table 5-232: por_cfgm_errgsr_mxp_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of MXP #{map[index%2]} <n> status	RO	64'h0

5.3.9.11 por_cfgm_errgsr_mxp_0-7_NS

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the XP <n> Non-secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3040 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-217: por_cfgm_errgsr_mxp_0-7_NS

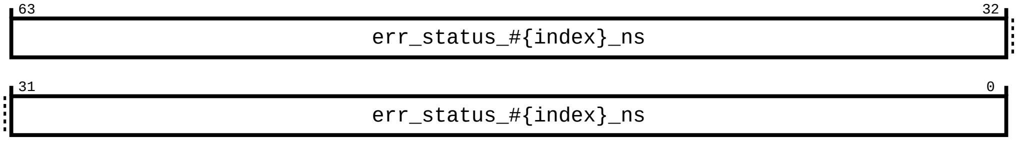


Table 5-233: por_cfgm_errgsr_mxp_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of MXP #{map[index%2]} <n> status	RO	64'h0

5.3.9.12 por_cfgm_errgsr_hni_0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the HN-I <n> Secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3080 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-218: por_cfgm_errgsr_hni_0-7

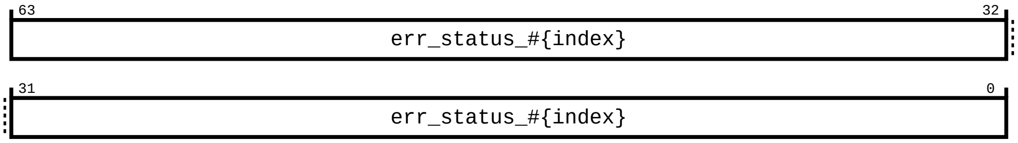


Table 5-234: por_cfgm_errgsr_hni_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of HN-I #{map[index%2]} <n> status	RO	64'h0

5.3.9.13 por_cfgm_errgsr_hni_0-7_NS

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the HN-I <n> Non-secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h30C0 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-219: por_cfgm_errgsr_hni_0-7_NS

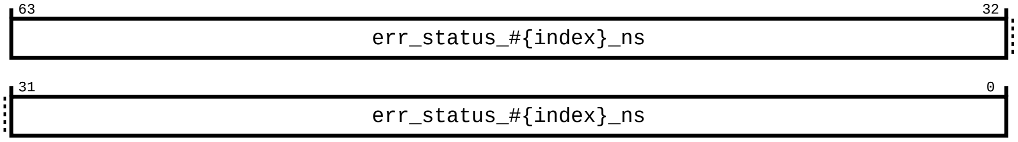


Table 5-235: por_cfgm_errgsr_hni_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of HN-I #{map[index%2]} <n> status	RO	64'h0

5.3.9.14 por_cfgm_errgsr_hnf_0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the HN-F <n> Secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3100 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-220: por_cfgm_errgsr_hnf_0-7

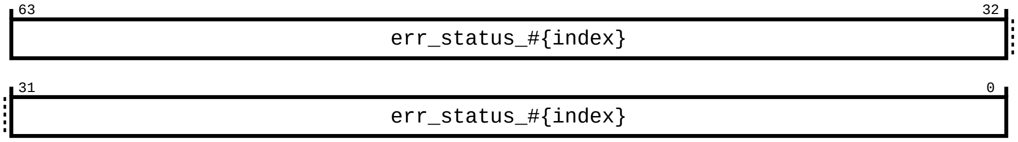


Table 5-236: por_cfgm_errgsr_hnf_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of HN-F #{map[index%2]} <n> status	RO	64'h0

5.3.9.15 por_cfgm_errgsr_hnf_0-7_NS

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the HN-F <n> Non-secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3140 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-221: por_cfgm_errgsr_hnf_0-7_NS

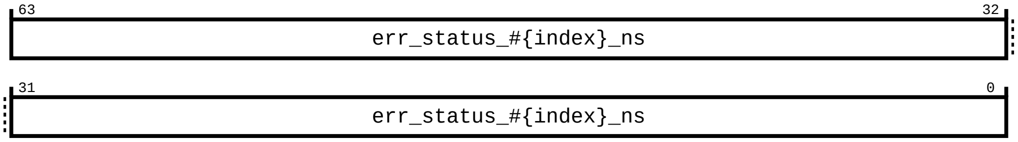


Table 5-237: por_cfgm_errgsr_hnf_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of HN-F #{map[index%2]} <n> status	RO	64'h0

5.3.9.16 por_cfgm_errgsr_sbsx_0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the SBSX <n> Secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3180 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-222: por_cfgm_errgsr_sbsx_0-7

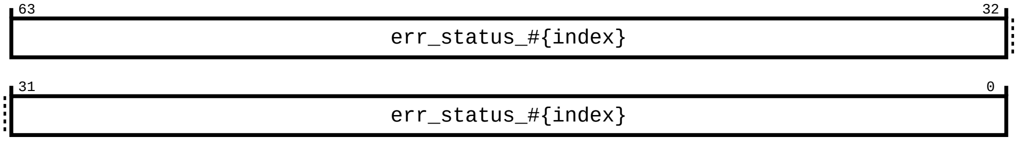


Table 5-238: por_cfgm_errgsr_sbsx_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of SBSX #{map[index%2]} <n> status	RO	64'h0

5.3.9.17 por_cfgm_errgsr_sbsx_0-7_NS

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the SBSX <n> Non-secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h31C0 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-223: por_cfgm_errgsr_sbsx_0-7_NS

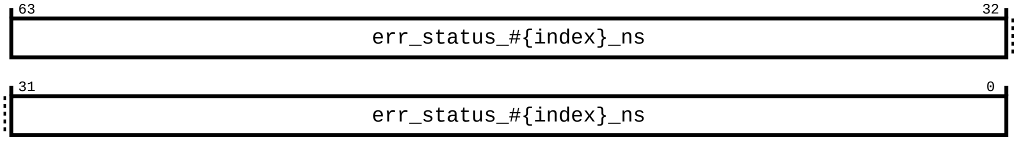


Table 5-239: por_cfgm_errgsr_sbsx_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of SBSX #{map[index%2]} <n> status	RO	64'h0

5.3.9.18 por_cfgm_errgsr_cxg_0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the CXG <n> Secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3200 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-224: por_cfgm_errgsr_cxg_0-7

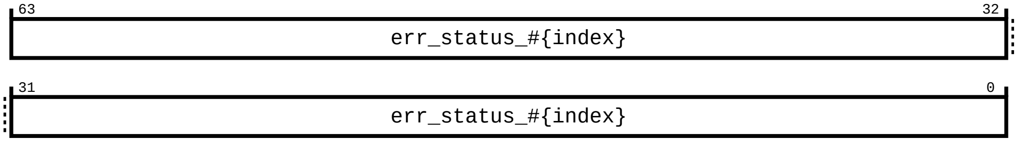


Table 5-240: por_cfgm_errgsr_cxg_0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}	Read-only copy of CXG #{map[index%2]} <n> status	RO	64'h0

5.3.9.19 por_cfgm_errgsr_cxg_0-7_NS

There are 8 iterations of this register, parameterized by the index from 0 to 7. Provides the CXG <n> Non-secure #{map[index%2]} status.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h3240 + (8 × #[0, 1, ... 7])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-225: por_cfgm_errgsr_cxg_0-7_NS

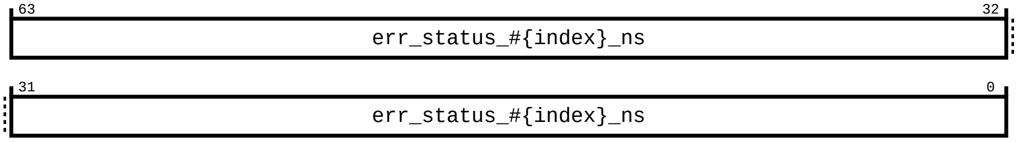


Table 5-241: por_cfgm_errgsr_cxg_0-7_NS attributes

Bits	Name	Description	Type	Reset
[63:0]	err_status_#{index}_ns	Read-only copy of CXG #{map[index%2]} <n> status	RO	64'h0

5.3.9.20 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

Figure 5-226: por_cfgm_errdevaff

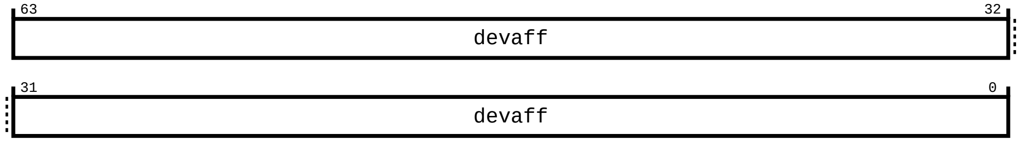


Table 5-242: por_cfgm_errdevaff attributes

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	64'b0

5.3.9.21 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

Figure 5-227: por_cfgm_errdevarch

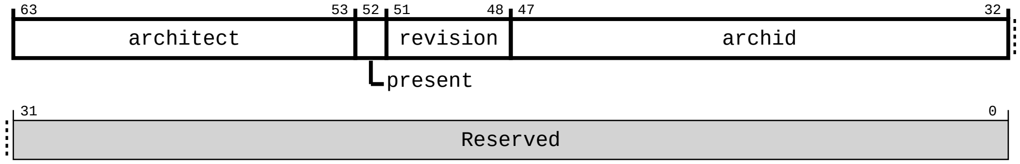


Table 5-243: por_cfgm_errdevarch attributes

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	-

5.3.9.22 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

Figure 5-228: por_cfgm_erridr

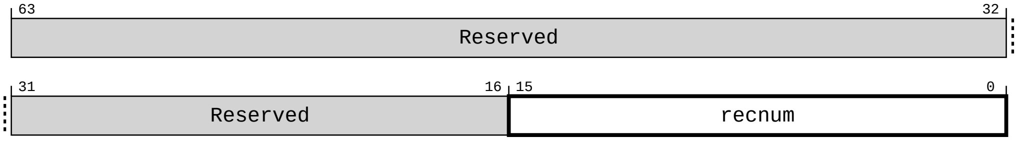


Table 5-244: por_cfgm_erridr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	recnum	Number of error records. This value is equal to 2 x (number of logical devices).	RO	Configuration dependent

5.3.9.23 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

Figure 5-229: por_cfgm_errpidr45

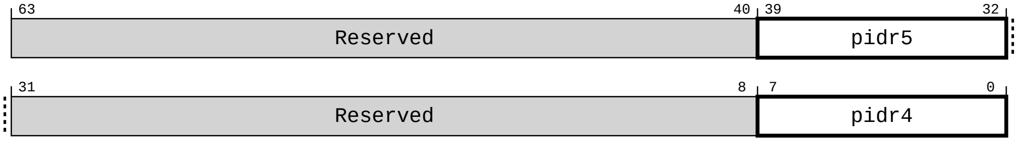


Table 5-245: por_cfgm_errpidr45 attributes

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

5.3.9.24 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

Figure 5-230: por_cfgm_errpidr67

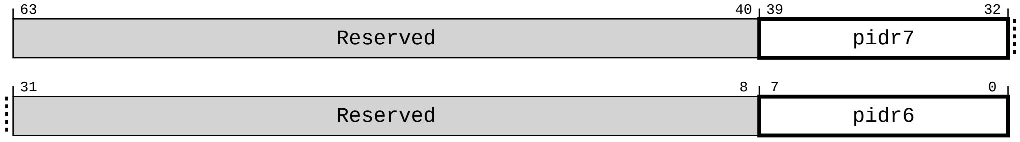


Table 5-246: por_cfgm_errpidr67 attributes

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

5.3.9.25 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

Figure 5-231: por_cfgm_errpidr01

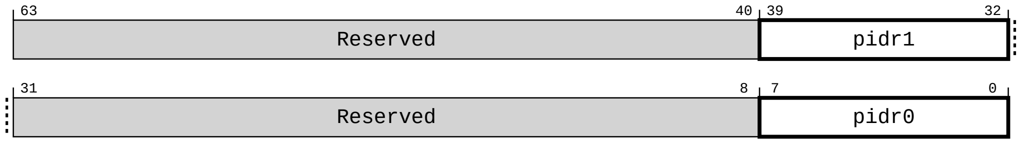


Table 5-247: por_cfgm_errpidr01 attributes

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

5.3.9.26 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

Figure 5-232: por_cfgm_errpidr23

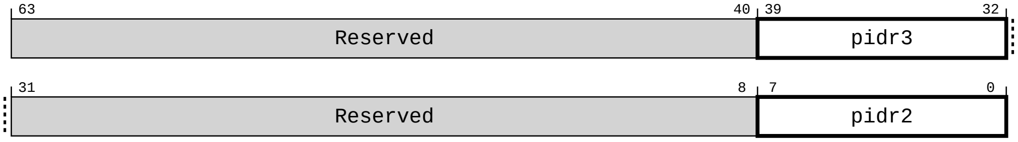


Table 5-248: por_cfgm_errpidr23 attributes

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

5.3.9.27 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

Figure 5-233: por_cfgm_errcidr01

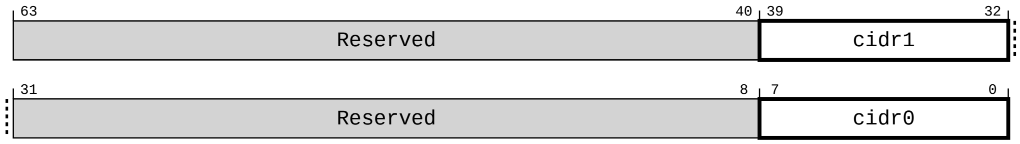


Table 5-249: por_cfgm_errcidr01 attributes

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

5.3.9.28 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

Figure 5-234: por_cfgm_errcidr23

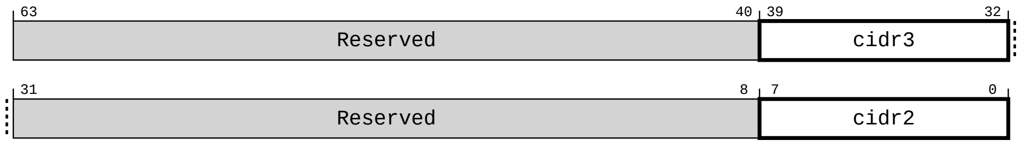


Table 5-250: por_cfgm_errcidr23 attributes

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

5.3.9.29 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

Figure 5-235: por_info_global

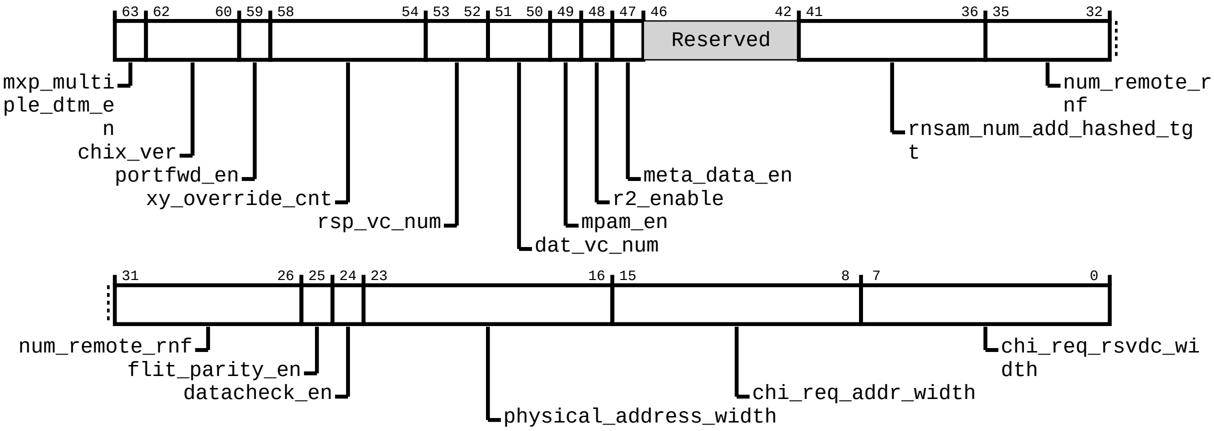


Table 5-251: por_info_global attributes

Bits	Name	Description	Type	Reset
[63]	mxp_multiple_dtm_en	Multiple DTMs feature enabled. This is used if the number of device ports on the XP is > 2	RO	Configuration dependent
[62:60]	chix_ver	CHI version parameter: 2 -> CHI-B, 3 -> CHI-C, 4 -> CHI-D, 5 -> CHI-E	RO	Configuration dependent
[59]	portfwd_en	CCIX port-to-port forwarding feature enabled	RO	Configuration dependent
[58:54]	xy_override_cnt	Number of source-target MXP pairs whose XY route path can be overridden	RO	Configuration dependent
[53:52]	rsp_vc_num	Number of extra RSP channels internal to MXP. For inreased bandwidth, this parameter must be set to 2.	RO	Configuration dependent
[51:50]	dat_vc_num	Number of extra DAT channels internal to MXP. For inreased bandwidth, this parameter must be set to 2.	RO	Configuration dependent
[49]	mpam_en	MPAM enabled	RO	Configuration dependent
[48]	r2_enable	CMN R2 features enabled	RO	Configuration dependent
[47]	meta_data_en	Metadata preservation mode enabled	RO	Configuration dependent
[46:42]	Reserved	Reserved	RO	-
[41:36]	rnsam_num_add_hashed_tgt	Number of extra hashed target IDs that are supported by the RN SAM, above the local HN-F 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 Data Check feature is enabled for CHI DAT flit	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[23:16]	physical_address_width	Physical Address (PA) width	RO	Configuration dependent
[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

5.3.9.30 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

Figure 5-236: por_ppu_int_enable

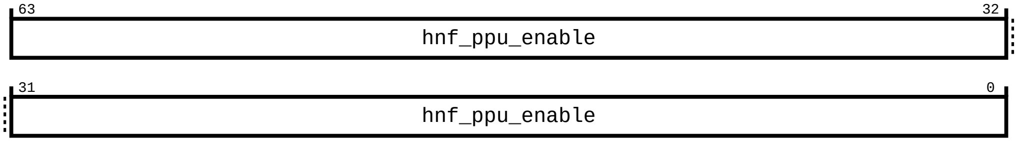


Table 5-252: por_ppu_int_enable attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_enable	Interrupt mask	RW	64'b0

5.3.9.31 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'h1C08

Type

W1C

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-237: por_ppu_int_status

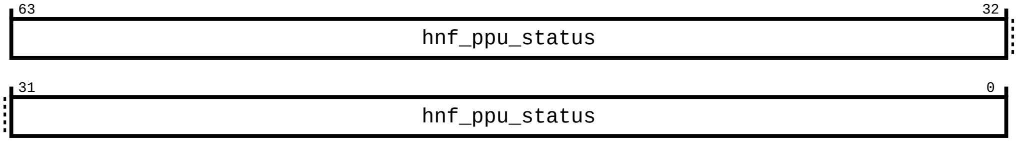


Table 5-253: por_ppu_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_status	Interrupt status	W1C	64'b0

5.3.9.32 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'h1C10

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-238: por_ppu_qactive_hyst

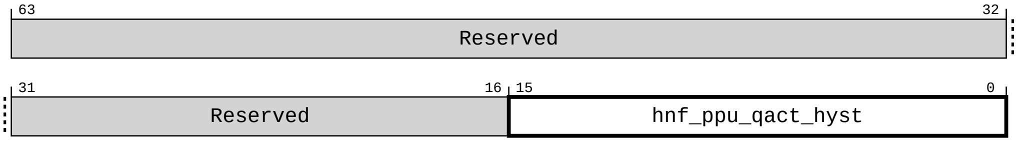


Table 5-254: por_ppu_qactive_hyst attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_ppu_qact_hyst	QACTIVE hysteresis	RW	16'h10

5.3.9.33 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'h1C18

Type

W1C

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-239: por_mpam_s_err_int_status

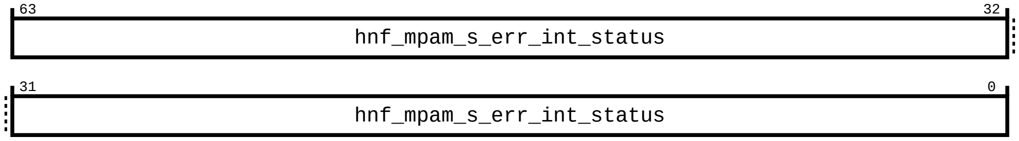


Table 5-255: por_mpam_s_err_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_s_err_int_status	MPAM Secure error interrupt status	W1C	64'b0

5.3.9.34 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'h1C20

Type

W1C

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-240: por_mpam_ns_err_int_status

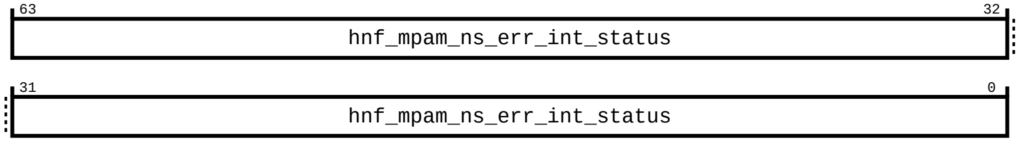


Table 5-256: por_mpam_ns_err_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_ns_err_int_status	MPAM Non-secure error interrupt status	W1C	64'b0

5.3.9.35 por_cfgm_child_pointer_0-255

There are 256 iterations of this register, parameterized by the index from 0 to 255. Contains the base address of a child configuration node.

Note

There are as many child pointer registers in the Global Config Unit as the number of XPs on the chip. Each successive child pointer register is at the next 8-byte address boundary. Each successive child pointer register is 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 × #[0, 1, ... 255])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-241: por_cfgm_child_pointer_0-255

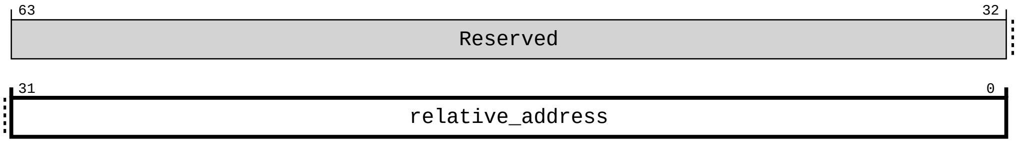


Table 5-257: por_cfgm_child_pointer_0-255 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	relative_address	Bit [31]: External or internal child node 1'b1: Indicates that the child pointer points to a configuration node that is external to CMN-650 1'b0: Indicates that the child pointer points to a configuration node that is internal to CMN-650 Bit [30]: Set to 1'b0 Bits [29:0]: Child node address offset relative to PERIPHBASE	RO	32'b0

5.3.10 Debug and trace register descriptions

This section lists the debug and trace registers.

5.3.10.1 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

Figure 5-242: por_dt_node_info

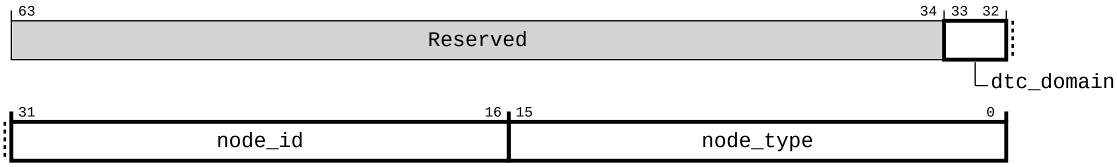


Table 5-258: por_dt_node_info attributes

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-650 node type identifier	RO	16'h3

5.3.10.2 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

Figure 5-243: por_dt_child_info

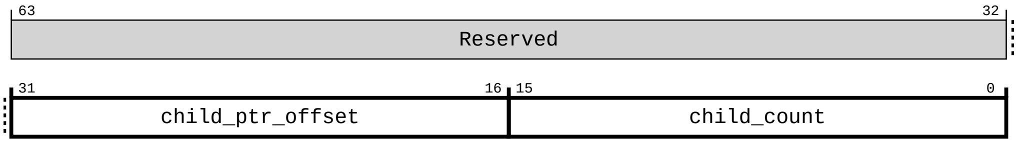


Table 5-259: por_dt_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.10.3 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

Figure 5-244: por_dt_secure_access

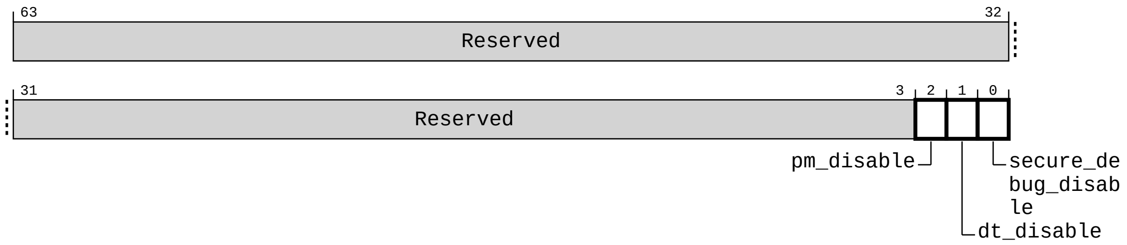


Table 5-260: por_dt_secure_access attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	pm_disable	Performance Monitoring Unit (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

Bits	Name	Description	Type	Reset
[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

5.3.10.4 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

Figure 5-245: por_dt_dtc_ctl

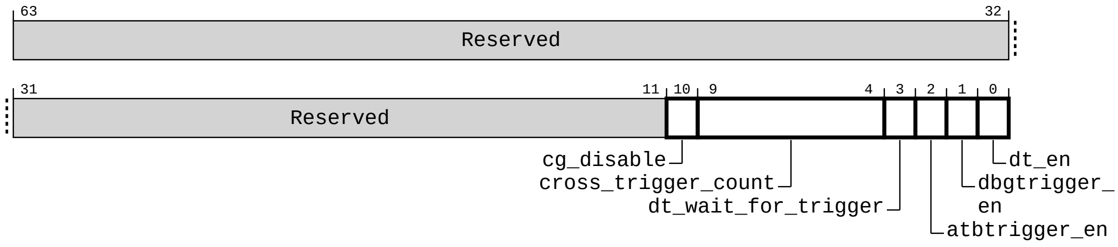


Table 5-261: por_dt_dtc_ctl attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	cg_disable	Disables DT architectural clock gates	RW	1'b0

Bits	Name	Description	Type	Reset
[9:4]	cross_trigger_count	Number of cross triggers received before trace enable NOTE: This field only applies 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 Performance Monitoring Unit (PMU) features	RW	1'b0

5.3.10.5 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

Figure 5-246: por_dt_trigger_status

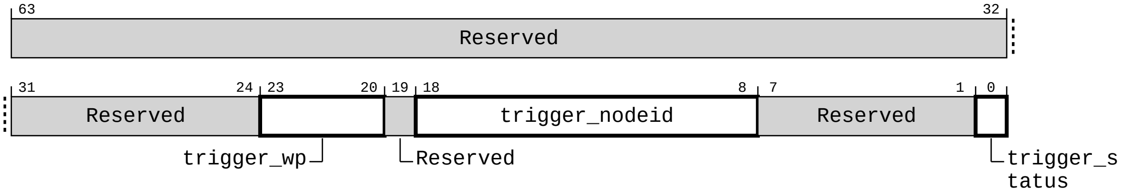


Table 5-262: por_dt_trigger_status attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:20]	trigger_wp	DBGWATCHTRIGREQ assertion, ATB trigger, or both DBGWATCHTRIGREQ assertion and ATB trigger are caused by watchpoint	RO	1'h0

Bits	Name	Description	Type	Reset
[19]	Reserved	Reserved	RO	-
[18:8]	trigger_nodeid	DBGWATCHTRIGREQ assertion, ATB trigger, or both DBGWATCHTRIGREQ assertion and ATB trigger are caused by node ID	RO	11'h0
[7:1]	Reserved	Reserved	RO	-
[0]	trigger_status	Indicates DBGWATCHTRIGREQ assertion, ATB trigger, or both DBGWATCHTRIGREQ assertion and ATB trigger	RO	1'h0

5.3.10.6 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

Figure 5-247: por_dt_trigger_status_clr

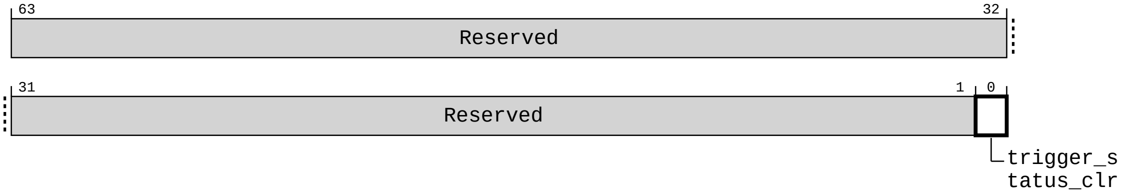


Table 5-263: por_dt_trigger_status_clr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	trigger_status_clr	Write a 1 to clear the trigger_status field of the por_dt_trigger_status register.	WO	1'b0

5.3.10.7 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

Figure 5-248: por_dt_trace_control

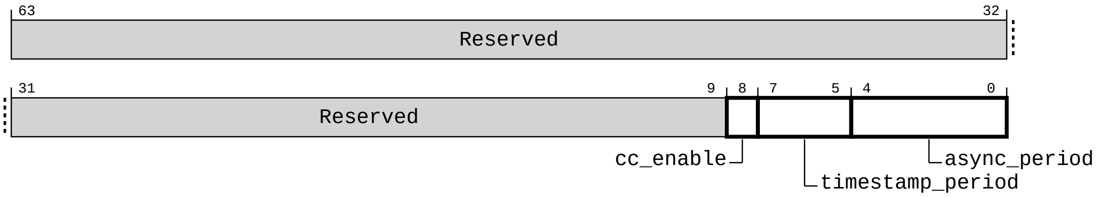


Table 5-264: por_dt_trace_control attributes

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

Bits	Name	Description	Type	Reset
[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 1,048,576B of trace Note: All other values are reserved.	RW	5'b0

5.3.10.8 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

Figure 5-249: por_dt_traceid

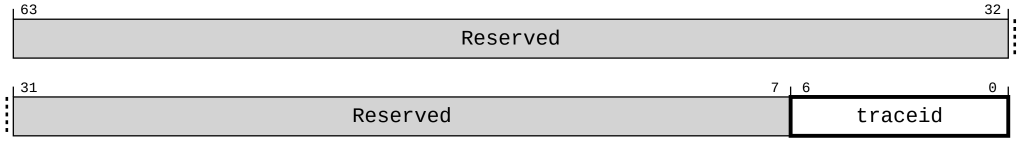


Table 5-265: por_dt_traceid attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[6:0]	traceid	ATB ID	RW	7'h0

5.3.10.9 por_dt_pmevcntAB

Contains the Performance Monitoring Unit (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

Figure 5-250: por_dt_pmevcntAB

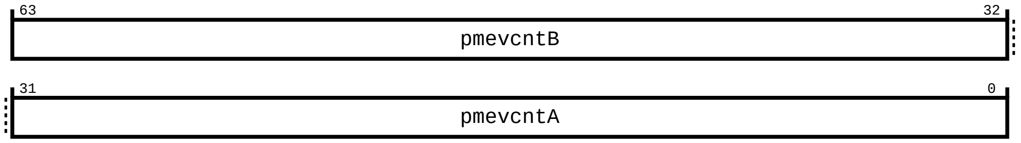


Table 5-266: por_dt_pmevcntAB attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntB	PMU counter B	RW	32'h0000
[31:0]	pmevcntA	PMU counter A	RW	32'h0000

5.3.10.10 por_dt_pmevcntCD

Contains the Performance Monitoring Unit (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

Figure 5-251: por_dt_pmevcntCD

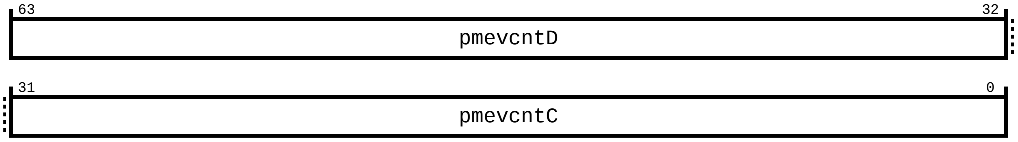


Table 5-267: por_dt_pmevcntCD attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntD	PMU counter D	RW	32'h0000
[31:0]	pmevcntC	PMU counter C	RW	32'h0000

5.3.10.11 por_dt_pmevcntEF

Contains the Performance Monitoring Unit (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

Figure 5-252: por_dt_pmevcntEF

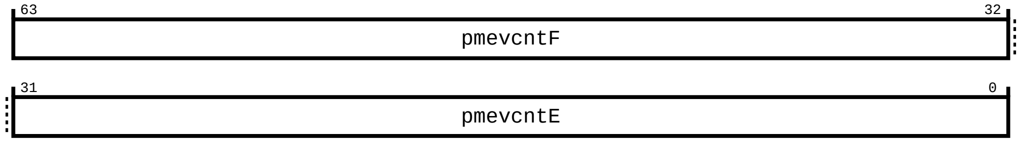


Table 5-268: por_dt_pmevcntEF attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntF	PMU counter F	RW	32'h0000
[31:0]	pmevcntE	PMU counter E	RW	32'h0000

5.3.10.12 por_dt_pmevcntGH

Contains the Performance Monitoring Unit (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

Figure 5-253: por_dt_pmevcntGH

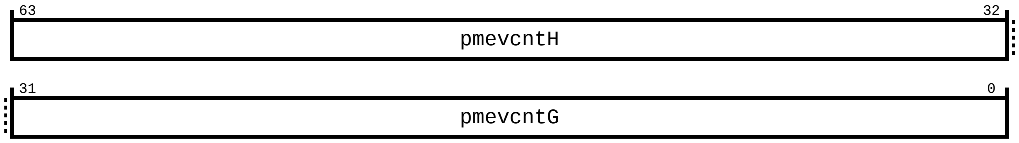


Table 5-269: por_dt_pmevcntGH attributes

Bits	Name	Description	Type	Reset
[63:32]	pmevcntH	PMU counter H	RW	32'h0000
[31:0]	pmevcntG	PMU counter G	RW	32'h0000

5.3.10.13 por_dt_pmccntr

Contains the Performance Monitoring Unit (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

Figure 5-254: por_dt_pmccntr

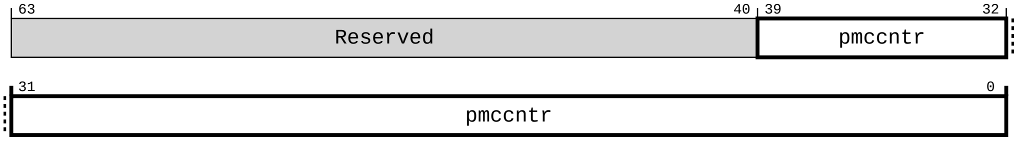


Table 5-270: por_dt_pmccntr attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:0]	pmccntr	PMU cycle counter	RW	40'h0

5.3.10.14 por_dt_pmevcntsrAB

Contains the Performance Monitoring Unit (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

Figure 5-255: por_dt_pmevcntsrAB

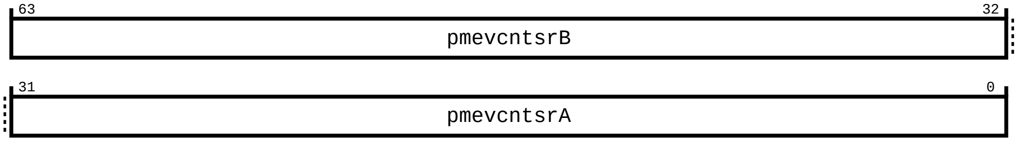


Table 5-271: por_dt_pmevcntsrAB attributes

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

5.3.10.15 por_dt_pmevcntsrCD

Contains the Performance Monitoring Unit (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

Figure 5-256: por_dt_pmevcntsrCD

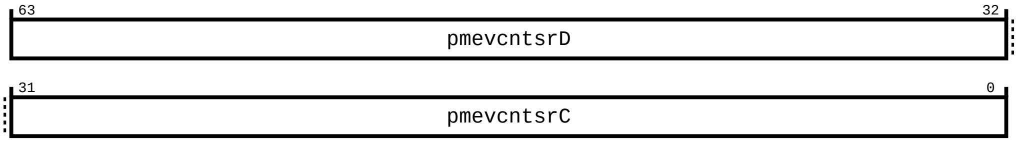


Table 5-272: por_dt_pmevcntsrCD attributes

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

5.3.10.16 por_dt_pmevcntsrEF

Contains the Performance Monitoring Unit (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

Figure 5-257: por_dt_pmevcntsrEF

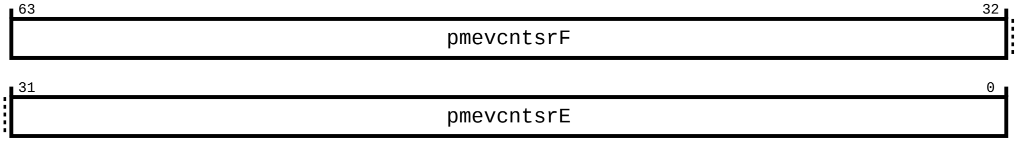


Table 5-273: por_dt_pmevcntsrEF attributes

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

5.3.10.17 por_dt_pmevcntsrGH

Contains the Performance Monitoring Unit (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

Figure 5-258: por_dt_pmevcntsrGH

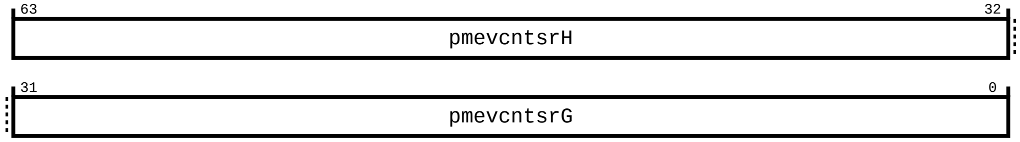


Table 5-274: por_dt_pmevcntsrGH attributes

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

5.3.10.18 por_dt_pmccntrsr

Contains the Performance Monitoring Unit (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

Figure 5-259: por_dt_pmccntrsr

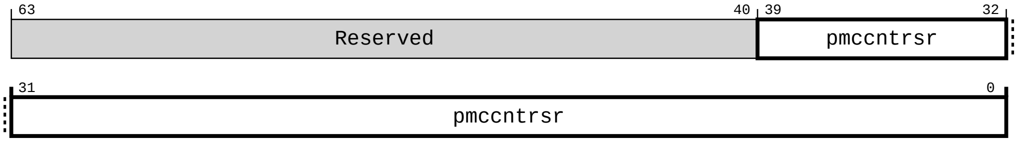


Table 5-275: por_dt_pmccntrsr attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:0]	pmccntrsr	PMU cycle counter shadow register	RW	40'h0

5.3.10.19 por_dt_pmcr

Functions as the Performance Monitoring Unit (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

Figure 5-260: por_dt_pmcr

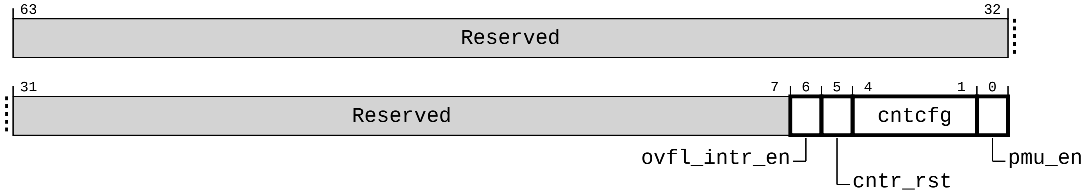


Table 5-276: por_dt_pmcr attributes

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 the ss_req field of the por_dt_pmsrr register 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

5.3.10.20 por_dt_pmovsr

Provides the Performance Monitoring Unit (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

Figure 5-261: por_dt_pmovsr

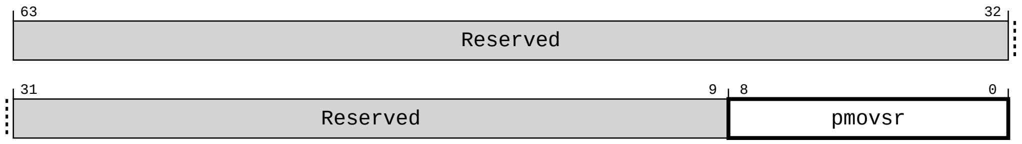


Table 5-277: por_dt_pmovsr attributes

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 7-0	RO	9'h0

5.3.10.21 por_dt_pmovsr_clr

Clears the Performance Monitoring Unit (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

Figure 5-262: por_dt_pmovsr_clr

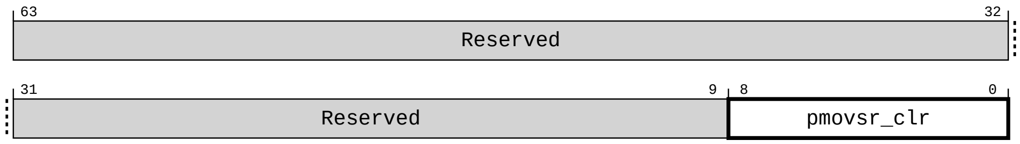


Table 5-278: por_dt_pmovsr_clr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8:0]	pmovsr_clr	Write a 1 to clear the corresponding pmovsr bit in the por_dt_pmovsr register.	WO	9'b0

5.3.10.22 por_dt_pmssr

Provides the Performance Monitoring Unit (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

Figure 5-263: por_dt_pmssr

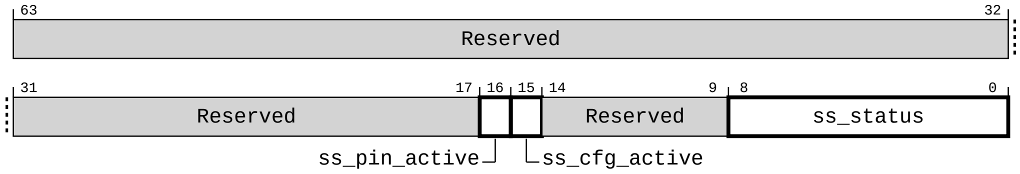


Table 5-279: por_dt_pmssr attributes

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	Activates PMU snapshot from configuration write	RO	1'b0
[14:9]	Reserved	Reserved	RO	-
[8:0]	ss_status	PMU snapshot status: Bit [8]: Indicates snapshot status for cycle counter Bits [7:0]: Indicates snapshot status for counters 7-0	RO	9'b0

5.3.10.23 por_dt_pmsrr

Sends Performance Monitoring Unit (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

Figure 5-264: por_dt_pmsrr

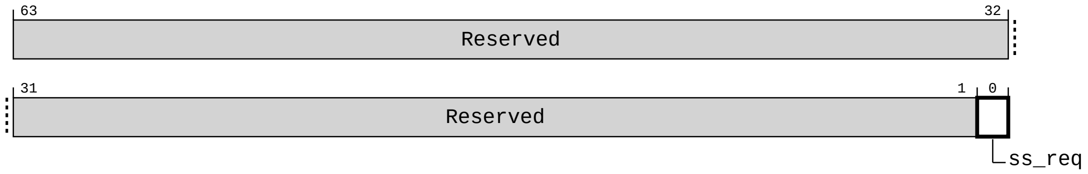


Table 5-280: por_dt_pmsrr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	ss_req	Write a 1 to request PMU snapshot.	WO	1'b0

5.3.10.24 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

Figure 5-265: por_dt_claim

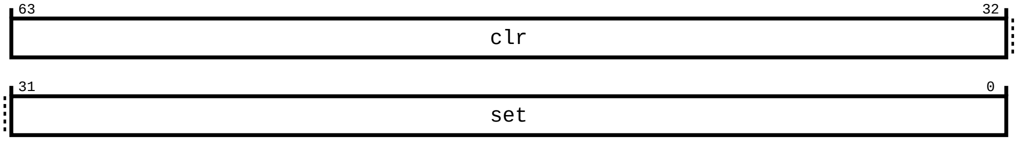


Table 5-281: por_dt_claim attributes

Bits	Name	Description	Type	Reset
[63:32]	clr	Upper half of the claim tag value. A write allows individual bits to be cleared and a read returns the current claim tag value.	RW	32'b0
[31:0]	set	Lower half of the claim tag value. A write allows individual bits to be set and a read returns the number of bits that can be set.	RW	32'hffffffff

5.3.10.25 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

Figure 5-266: por_dt_devaff

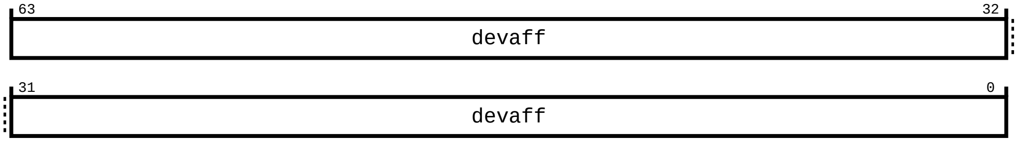


Table 5-282: por_dt_devaff attributes

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	64'b0

5.3.10.26 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

Figure 5-267: por_dt_lsr

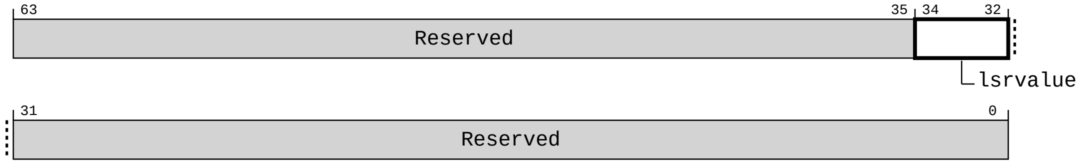


Table 5-283: por_dt_lsr attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34:32]	lsrvalue	Lock status value	RO	3'b0
[31:0]	Reserved	Reserved	RO	-

5.3.10.27 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

Figure 5-268: por_dt_authstatus_devarch

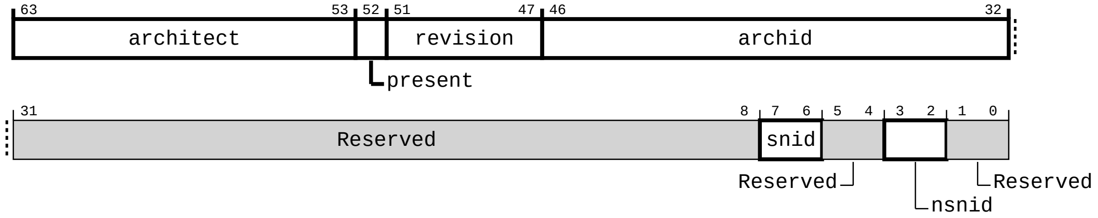


Table 5-284: por_dt_authstatus_devarch attributes

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	-

5.3.10.28 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

Figure 5-269: por_dt_devid

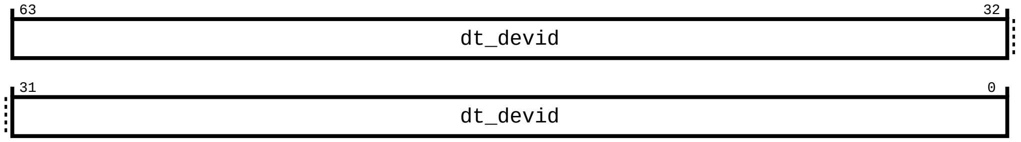


Table 5-285: por_dt_devid attributes

Bits	Name	Description	Type	Reset
[63:0]	dt_devid	Device ID	RO	64'b0

5.3.10.29 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

Figure 5-270: por_dt_devtype

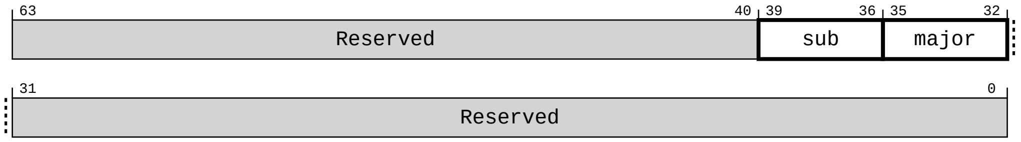


Table 5-286: por_dt_devtype attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:36]	sub	Subtype	RO	4'h4
[35:32]	major	Major type	RO	4'h3
[31:0]	Reserved	Reserved	RO	-

5.3.10.30 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

Figure 5-271: por_dt_pidr45

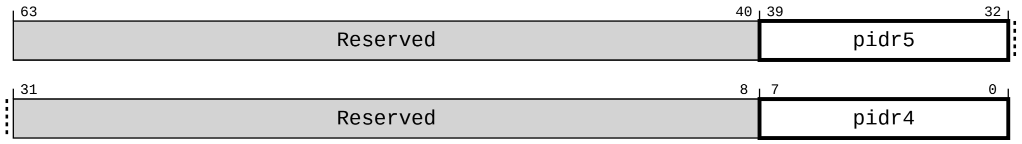


Table 5-287: por_dt_pidr45 attributes

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

5.3.10.31 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

Figure 5-272: por_dt_pidr67

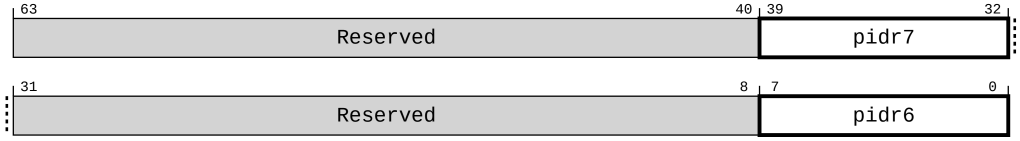


Table 5-288: por_dt_pidr67 attributes

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

5.3.10.32 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

Figure 5-273: por_dt_pidr01

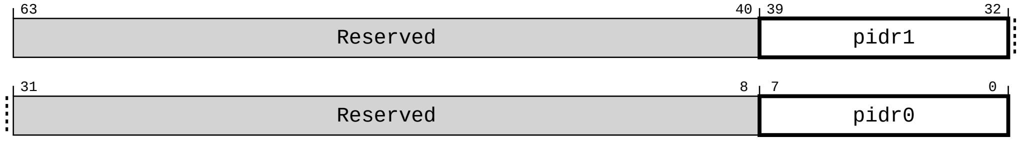


Table 5-289: por_dt_pidr01 attributes

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

5.3.10.33 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

Figure 5-274: por_dt_pidr23

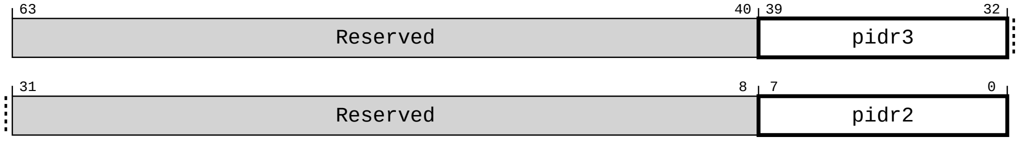


Table 5-290: por_dt_pidr23 attributes

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

5.3.10.34 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

Figure 5-275: por_dt_cidr01

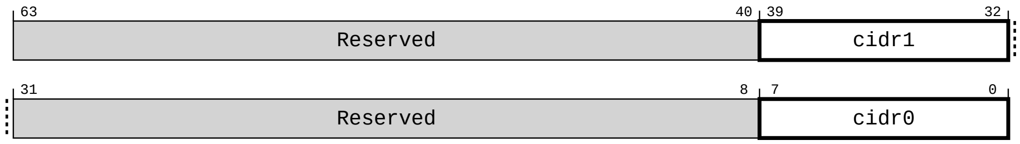


Table 5-291: por_dt_cidr01 attributes

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

5.3.10.35 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

Figure 5-276: por_dt_cidr23

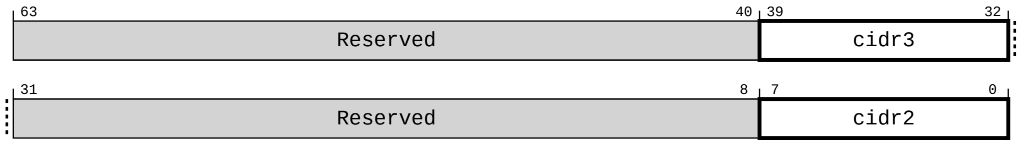


Table 5-292: por_dt_cidr23 attributes

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

5.3.11 SBSX register descriptions

This section lists the SBSX registers.

5.3.11.1 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

Figure 5-277: por_sbsx_node_info

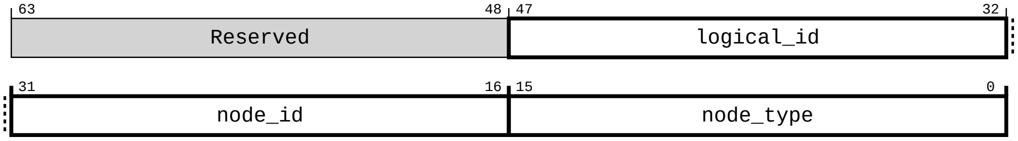


Table 5-293: por_sbsx_node_info attributes

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-650 node type identifier	RO	16'h0007

5.3.11.2 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

Figure 5-278: por_sbsx_child_info

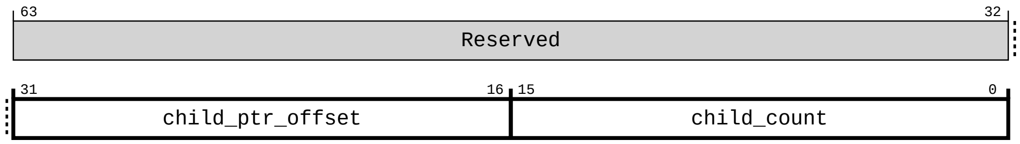


Table 5-294: por_sbsx_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.11.3 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

Figure 5-279: por_sbsx_secure_register_groups_override

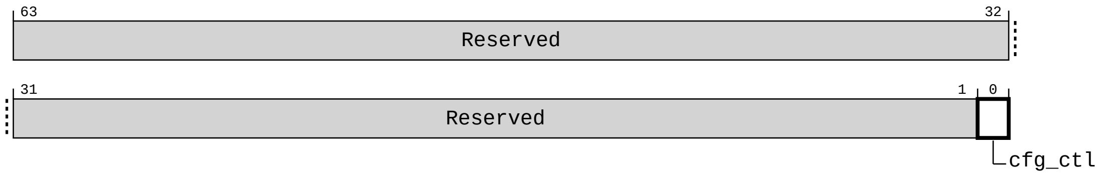


Table 5-295: por_sbsx_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	cfg_ctl	Allows Non-secure access to Secure configuration control register, por_sbsx_cfg_ctl	RW	1'b0

5.3.11.4 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

Figure 5-280: por_sbsx_unit_info

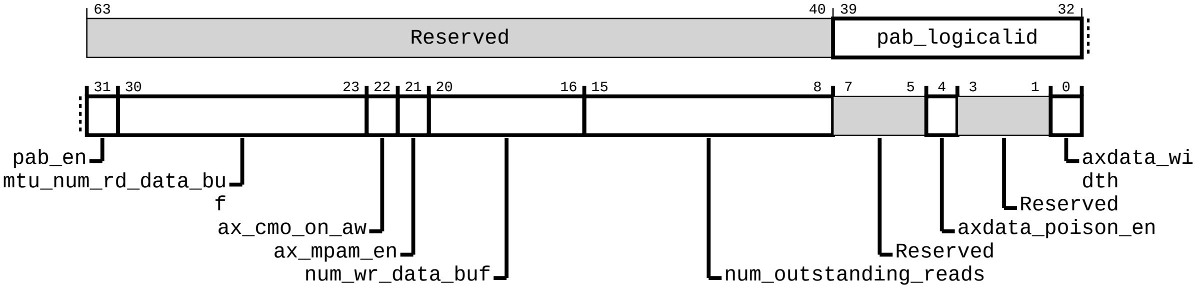


Table 5-296: por_sbsx_unit_info attributes

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 AXI/ACE-Lite interface: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[21]	ax_mpam_en	MPAM enable on AXI/ACE-Lite 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 AXI/ACE-Lite interface: 1'b0: Not supported 1'b1: Supported	RO	Configuration dependent
[3:1]	Reserved	Reserved	RO	-
[0]	axdata_width	Data width on AXI/ACE-Lite interface: 1'b0: 128 bits 1'b1: 256 bits	RO	Configuration dependent

5.3.11.5 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

Figure 5-281: por_sbsx_cfg_ctl

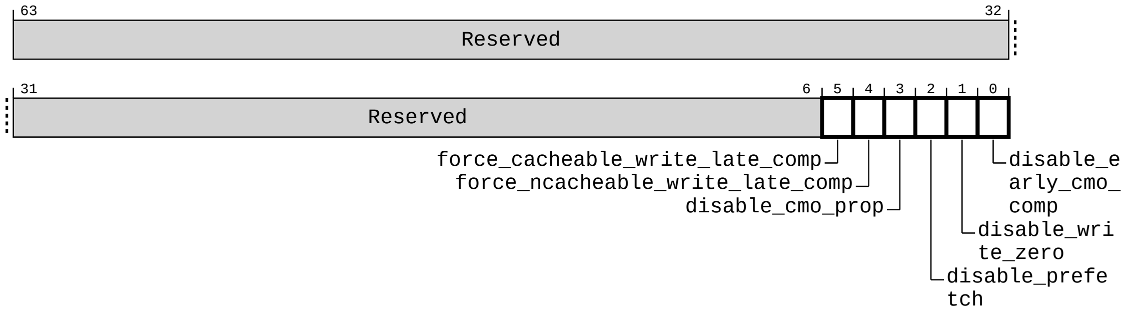


Table 5-297: por_sbsx_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5]	force_cacheable_write_late_comp	Late Comp for Cacheable writes. Overrides EWA.	RW	1'b0
[4]	force_ncacheable_write_late_comp	Late Comp for Non-cacheable writes. Overrides EWA.	RW	1'b0
[3]	disable_cmo_prop	Disables CMO propagation on ACE	RW	1'b0

Bits	Name	Description	Type	Reset
[2]	disable_prefetch	Disables prefetches on AXI	RW	1'b0
[1]	disable_write_zero	Disables WriteZero operation on AXI	RW	1'b0
[0]	disable_early_cmo_comp	Disables early Comp for CMOs in SBSX to HN-F	RW	1'b0

5.3.11.6 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

Figure 5-282: por_sbsx_aux_ctl

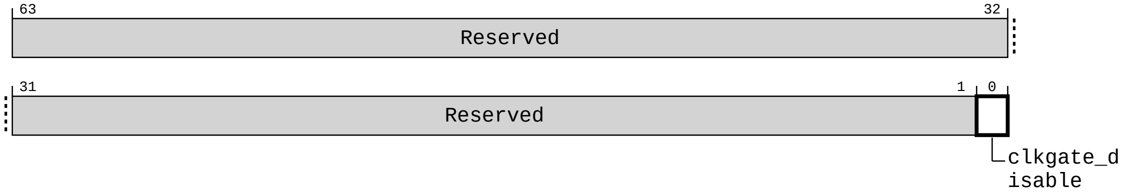


Table 5-298: por_sbsx_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	clkgate_disable	Disables internal clock gating in SBSX bridge	RW	1'b0

5.3.11.7 por_sbsx_cbusy_limit_ctl

Completer Busy (CBusy) threshold limits for Request Tracker (ReqTracker) 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

Figure 5-283: por_sbsx_cbusy_limit_ctl

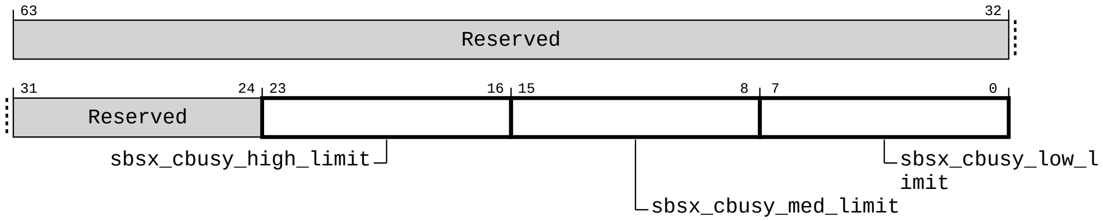


Table 5-299: por_sbsx_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	sbsx_cbusy_high_limit	ReqTracker limit for CBusy high	RW	Configuration dependent
[15:8]	sbsx_cbusy_med_limit	ReqTracker limit for CBusy medium	RW	Configuration dependent
[7:0]	sbsx_cbusy_low_limit	ReqTracker limit for CBusy low	RW	Configuration dependent

5.3.11.8 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-284: por_sbsx_errfr

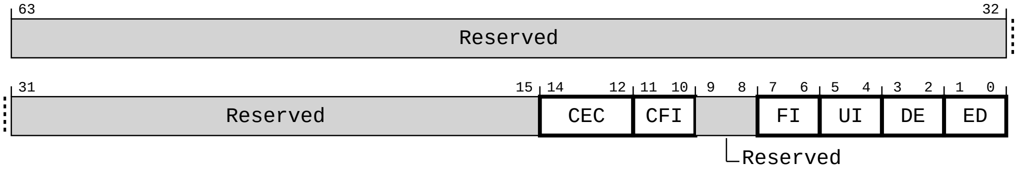


Table 5-300: por_sbsx_errfr attributes

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
[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

5.3.11.9 por_sbsx_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-285: por_sbsx_errctlr

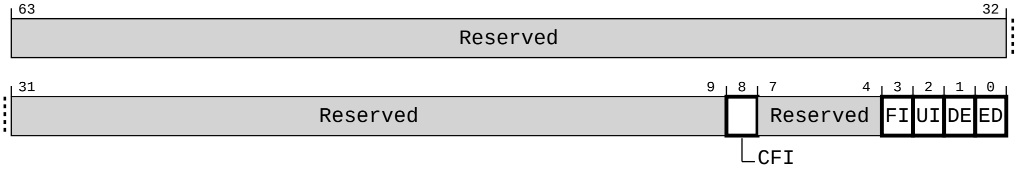


Table 5-301: por_sbsx_errctlr attributes

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

5.3.11.10 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-286: por_sbsx_errstatus

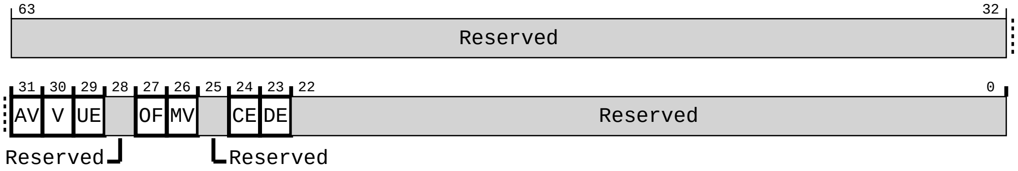


Table 5-302: por_sbsx_errstatus attributes

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

Bits	Name	Description	Type	Reset
[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 slave 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, and 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	-
[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	-

5.3.11.11 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-287: por_sbsx_erraddr

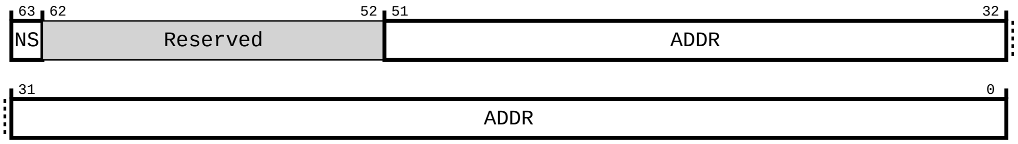


Table 5-303: por_sbsx_erraddr attributes

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

5.3.11.12 por_sbsx_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-288: por_sbsx_errmisc

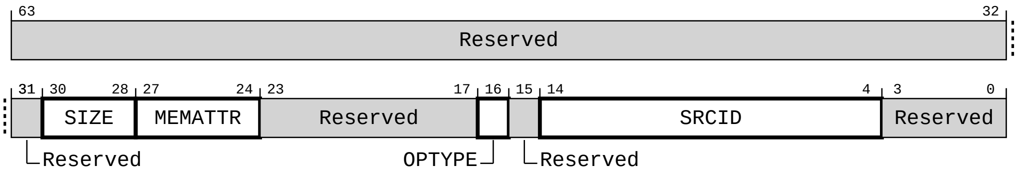


Table 5-304: por_sbsx_errmisc attributes

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	-

5.3.11.13 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

Figure 5-289: por_sbsx_errfr_NS

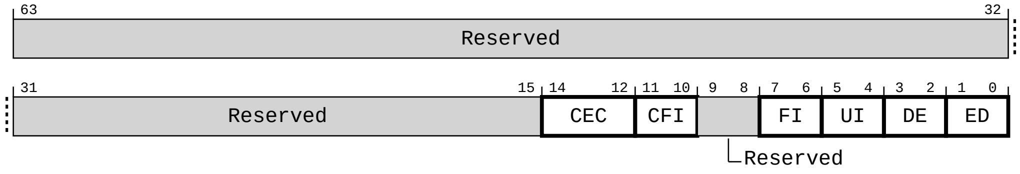


Table 5-305: por_sbsx_errfr_NS attributes

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

5.3.11.14 por_sbsx_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection and 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

Figure 5-290: por_sbsx_errctlr_NS

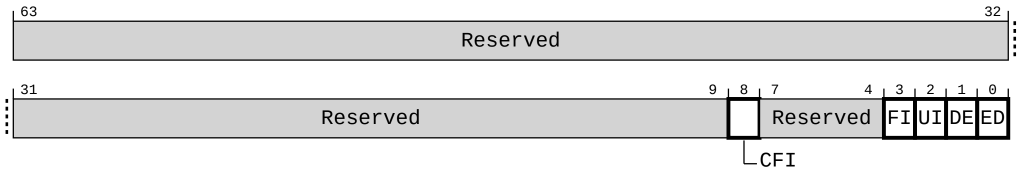


Table 5-306: por_sbsx_errctlr_NS attributes

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

5.3.11.15 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

Figure 5-291: por_sbsx_errstatus_NS

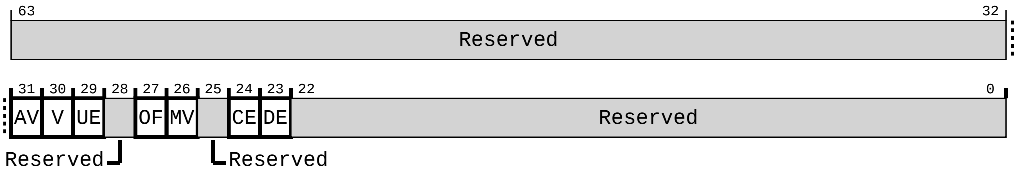


Table 5-307: por_sbsx_errstatus_NS attributes

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 slave 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, and CE fields	W1C	1'b0
[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	-

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	-

5.3.11.16 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

Figure 5-292: por_sbsx_erraddr_NS

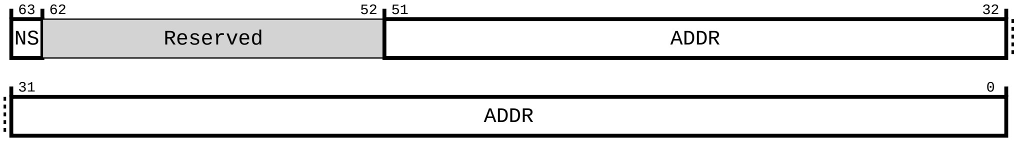


Table 5-308: por_sbsx_erraddr_NS attributes

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

5.3.11.17 por_sbsx_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred and 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

Figure 5-293: por_sbsx_errmisc_NS

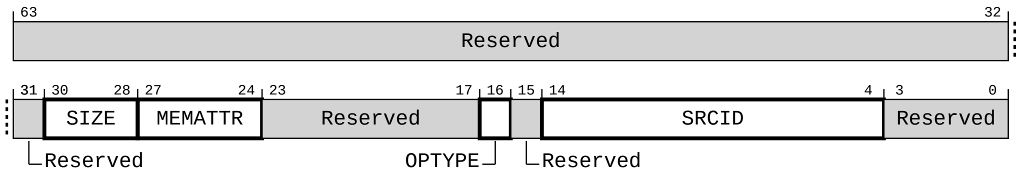


Table 5-309: por_sbsx_errmisc_NS attributes

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	-

5.3.11.18 por_sbsx_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-294: por_sbsx_pmu_event_sel

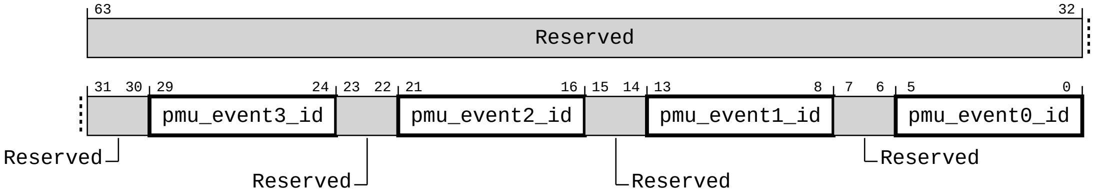


Table 5-310: por_sbsx_pmu_event_sel attributes

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	-

Bits	Name	Description	Type	Reset
[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, in other words TXDAT valid but no link credit available 6'h25: TXRSP stall, in other words TXRSP valid but no link credit available Note: All other encodings are reserved.	RW	6'b0

5.3.12 HN-F MPAM_NS register descriptions

This section lists the HN-F MPAM_NS registers.

5.3.12.1 por_hnf_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

Figure 5-295: por_hnf_mpam_ns_node_info

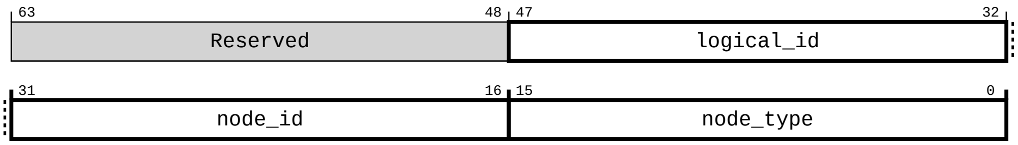


Table 5-311: por_hnf_mpam_ns_node_info attributes

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-650 node type identifier	RO	16'h0009

5.3.12.2 por_hnf_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

Figure 5-296: por_hnf_mpam_ns_child_info

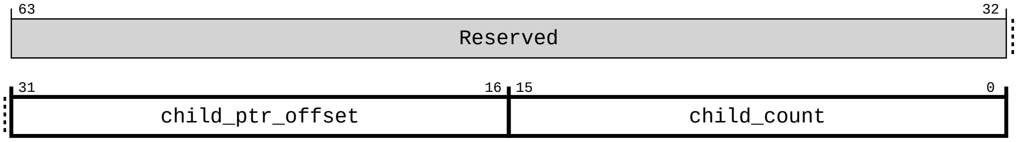


Table 5-312: por_hnf_mpam_ns_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.12.3 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-297: por_hnf_mpam_idr

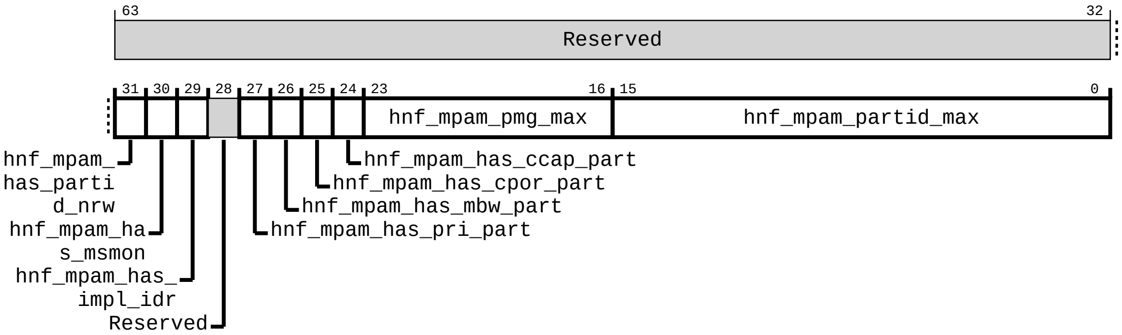


Table 5-313: por_hnf_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_mpam_has_partid_nrw	0: HN-F does not support MPAM PARTID narrowing 1: HN-F supports MPAM PARTID narrowing	RO	0
[30]	hnf_mpam_has_msmon	0: MPAM performance monitoring is not supported 1: MPAM performance monitoring is supported	RO	1
[29]	hnf_mpam_has_impl_idr	0: MPAM implementation specific partitioning features not supported 1: MPAM implementation specific partitioning features supported	RO	0
[28]	Reserved	Reserved	RO	-
[27]	hnf_mpam_has_pri_part	0: MPAM priority partitioning is not supported 1: MPAM priority partitioning is supported	RO	0
[26]	hnf_mpam_has_mbw_part	0: MPAM memory bandwidth partitioning is not supported 1: MPAM memory bandwidth partitioning is supported	RO	0
[25]	hnf_mpam_has_cpor_part	0: MPAM cache portion partitioning is not supported 1: MPAM cache portion partitioning is supported	RO	1
[24]	hnf_mpam_has_ccap_part	0: MPAM cache maximum capacity partitioning is not supported 1: MPAM cache maximum capacity partitioning is supported	RO	1
[23:16]	hnf_mpam_pmg_max	Maximum value of Non-secure PMG supported by this HN-F	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[15:0]	hnf_mpam_partid_max	Maximum value of Non-secure PARTID supported by this HN-F	RO	Configuration dependent

5.3.12.4 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-298: por_hnf_mpam_iidr

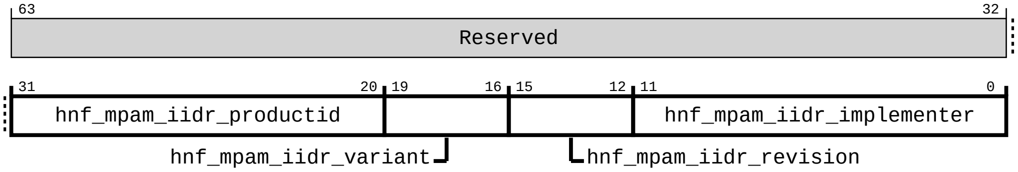


Table 5-314: por_hnf_mpam_iidr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hnf_mpam_iidr_productid	Implementation-defined value identifying the MPAM memory system component	RO	12'h000
[19:16]	hnf_mpam_iidr_variant	Implementation-defined value identifying the major revision of the product	RO	4'b0000
[15:12]	hnf_mpam_iidr_revision	Implementation-defined value identifying the minor revision of the product	RO	4'b0000
[11:0]	hnf_mpam_iidr_implementer	Implementation-defined value identifying the company that implemented MPAM memory system component	RO	12'h43B

5.3.12.5 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-299: por_hnf_mpam_aidr

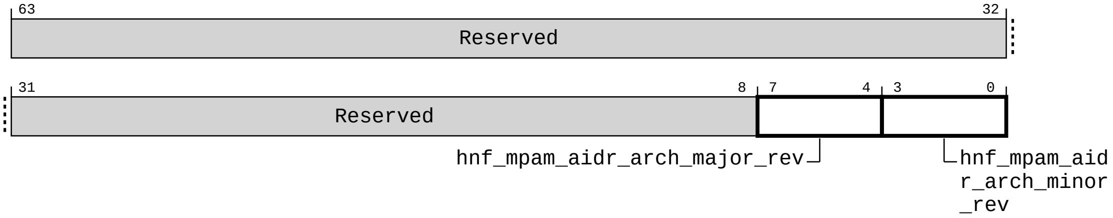


Table 5-315: por_hnf_mpam_aidr attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hnf_mpam_aidr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	4'b0001
[3:0]	hnf_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	4'b0000

5.3.12.6 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-300: por_hnf_mpam_impl_idr

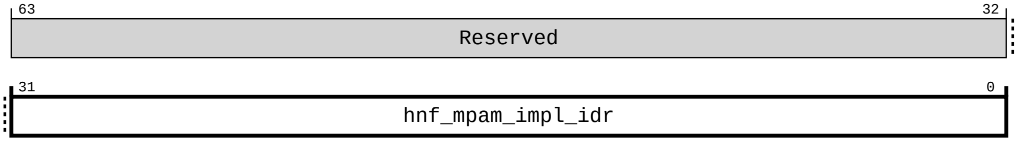


Table 5-316: por_hnf_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hnf_mpam_impl_idr	Implementation-defined partitioning features.	RO	32'h00000000

5.3.12.7 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-301: por_hnf_mpam_cpor_idr

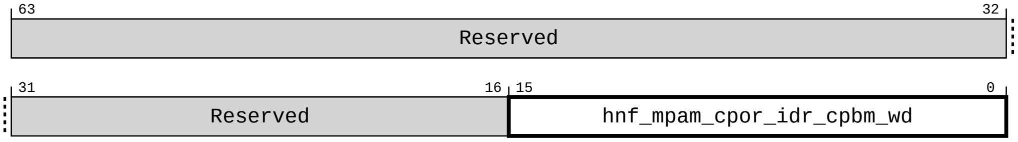


Table 5-317: por_hnf_mpam_cpor_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bitmap of this device.	RO	Configuration dependent

5.3.12.8 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-302: por_hnf_mpam_ccap_idr

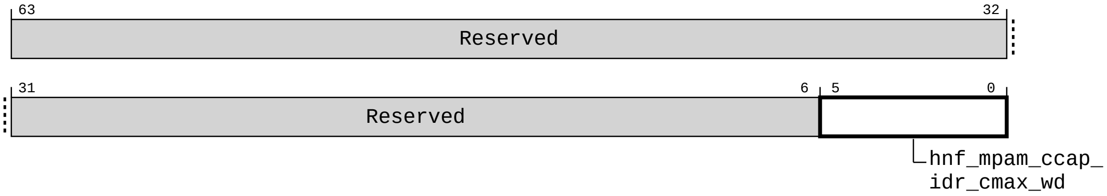


Table 5-318: por_hnf_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[5:0]	hnf_mpam_ccap_idr_cmax_wd	Number of fractional bits that are implemented in the cache capacity partitioning	RO	Configuration dependent

5.3.12.9 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-303: por_hnf_mpam_mbw_idr

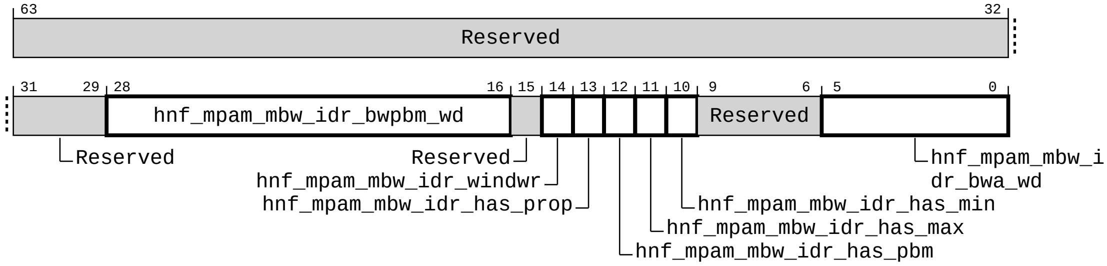


Table 5-319: por_hnf_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hnf_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hnf_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]	hnf_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]	hnf_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]	hnf_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]	hnf_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]	hnf_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

5.3.12.10 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-304: por_hnf_mpam_pri_idr

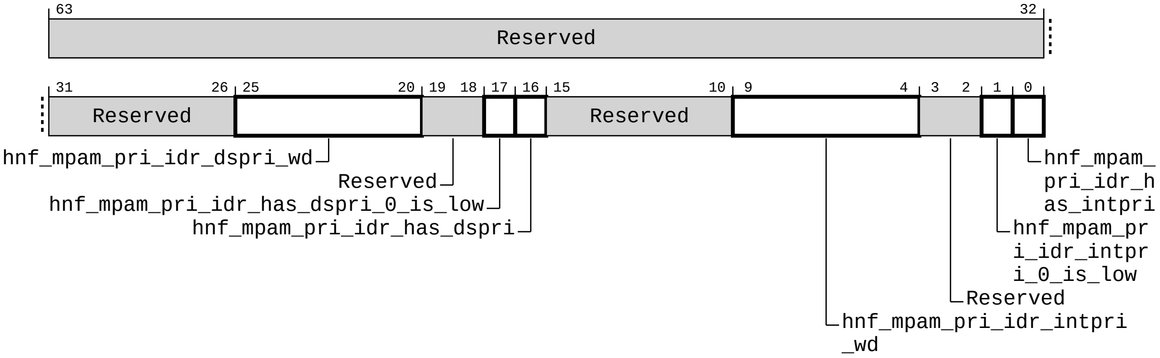

63
32
Reserved
31
26
25
20
19
18
17
16
15
10
9
4
3
2
1
0
Reserved
Reserved
hnf_mpam_pri_idr_dspri_wd
Reserved
hnf_mpam_pri_idr_has_dspri_0_is_low
hnf_mpam_pri_idr_has_dspri
hnf_mpam_pri_idr_has_intpri
hnf_mpam_pri_idr_intpri_0_is_low
Reserved
hnf_mpam_pri_idr_intpri_wd

Table 5-320: por_hnf_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	hnf_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]	hnf_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]	hnf_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]	hnf_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]	hnf_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]	hnf_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

5.3.12.11 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-305: por_hnf_mpam_partid_nrw_idr

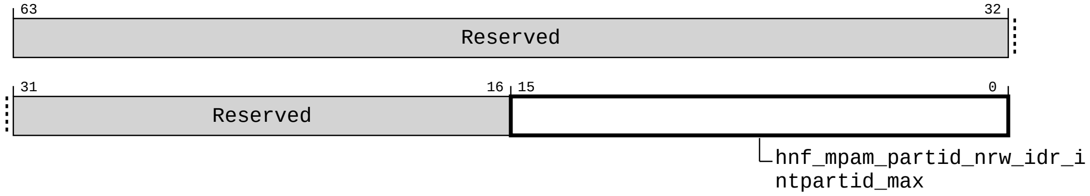


Table 5-321: por_hnf_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

5.3.12.12 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-306: por_hnf_mpam_msmon_idr

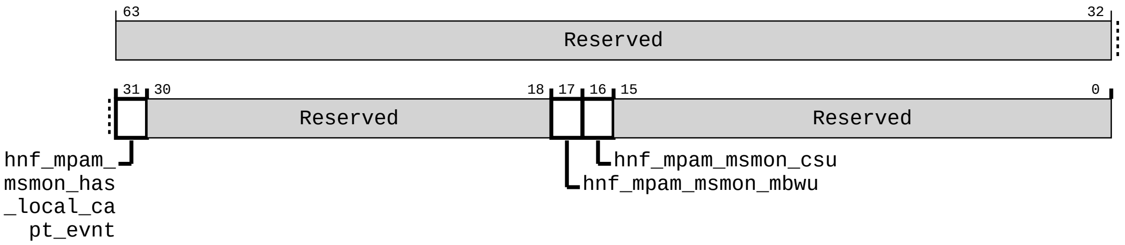


Table 5-322: por_hnf_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_mpam_msmon_mbwu	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	hnf_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	-

5.3.12.13 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-307: por_hnf_mpam_csumon_idr

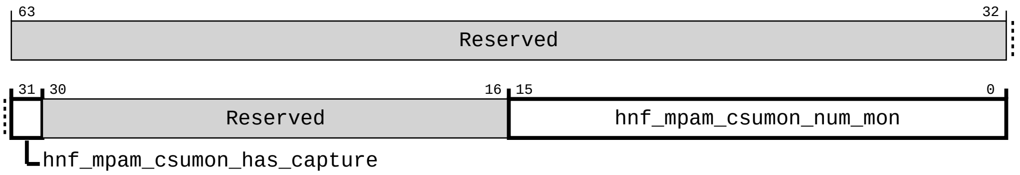


Table 5-323: por_hnf_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_mpam_csumon_has_capture	0: MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in the CSU monitor feature of this component. 1: The CSU monitor feature of this component has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behavior.	RO	1'h1
[30:16]	Reserved	Reserved	RO	-
[15:0]	hnf_mpam_csumon_num_mon	The number of CSU monitoring counters that are implemented in this component.	RO	Configuration dependent

5.3.12.14 por_hnf_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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-308: por_hnf_mpam_mbwumon_idr

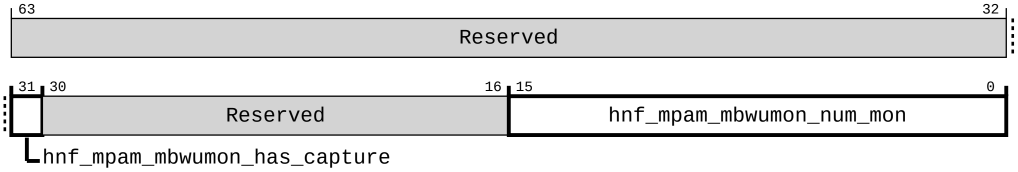


Table 5-324: por_hnf_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_mpam_mbwumon_has_capture	0: MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in the MBWU monitor feature of this component. 1: The MBWU monitor feature of this component has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behavior.	RO	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:0]	hnf_mpam_mbwumon_num_mon	The number of MBWU monitoring counters that are implemented in this component.	RO	16'h0

5.3.12.15 por_hnf_ns_mpam_ecr

MPAM Error Control Register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-309: por_hnf_ns_mpam_ecr

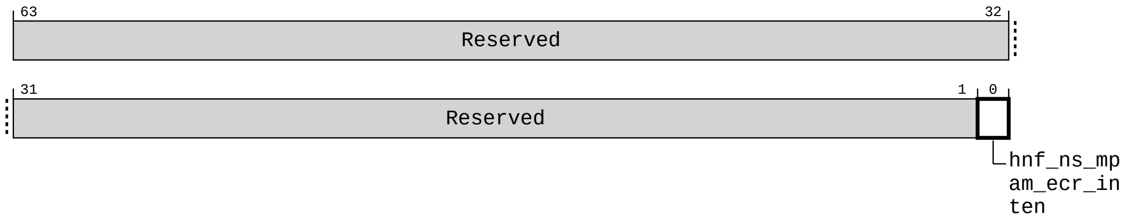


Table 5-325: por_hnf_ns_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hnf_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

5.3.12.16 por_hnf_ns_mpam_esr

MPAM Error Status Register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-310: por_hnf_ns_mpam_esr

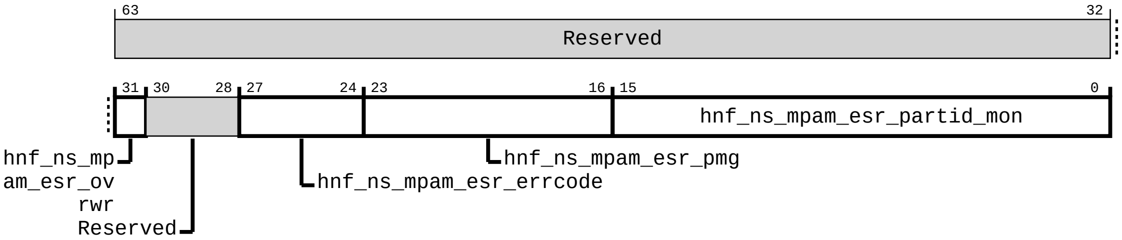


Table 5-326: por_hnf_ns_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_ns_mpam_esr_errcode	Error code	RW	4'h0
[23:16]	hnf_ns_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hnf_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

5.3.12.17 por_hnf_ns_mpamcfg_part_sel

MPAM partition configuration selection register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-311: por_hnf_ns_mpamcfg_part_sel

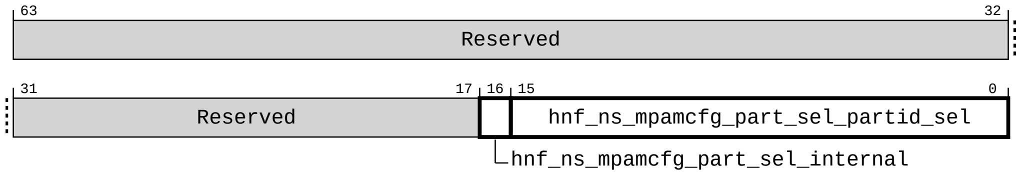

63
32
Reserved
31
17
16
15
0
Reserved
hnf_ns_mpamcfg_part_sel_partid_sel
hnf_ns_mpamcfg_part_sel_internal

Table 5-327: por_hnf_ns_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hnf_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]	hnf_ns_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

5.3.12.18 por_hnf_ns_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register. This register is banked seperately for S and NS

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-312: por_hnf_ns_mpamcfg_cmax

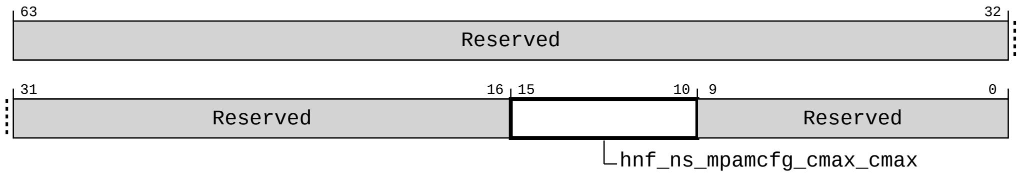


Table 5-328: por_hnf_ns_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:10]	hnf_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	6'h3f
[9:0]	Reserved	Reserved	RO	-

5.3.12.19 por_hnf_ns_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-313: por_hnf_ns_mpamcfg_mbw_min

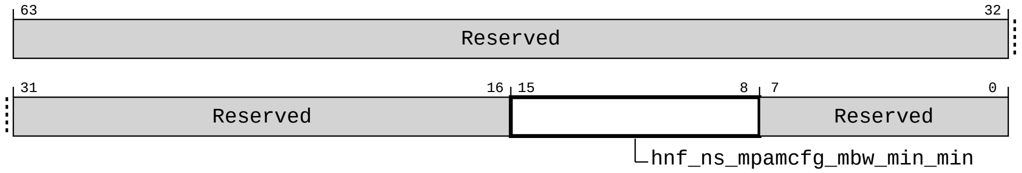


Table 5-329: por_hnf_ns_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	hnf_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	-

5.3.12.20 por_hnf_ns_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked seperately for S and 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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-314: por_hnf_ns_mpamcfg_mbw_max

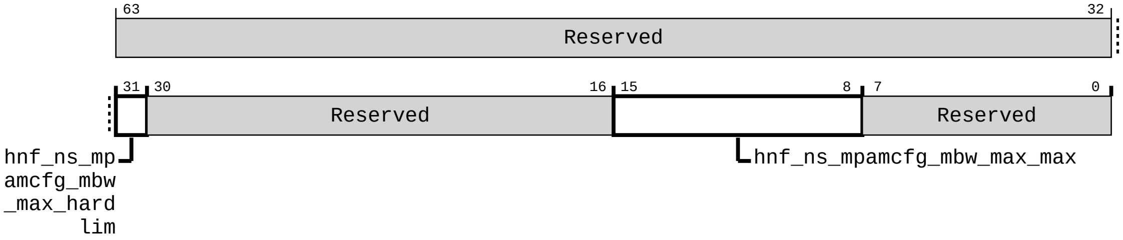


Table 5-330: por_hnf_ns_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_ns_mpamcfg_mbw_max_hardlim	0: When maximum bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth. 1: When maximum 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]	hnf_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	-

5.3.12.21 por_hnf_ns_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register. This register is banked seperately for S and 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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-315: por_hnf_ns_mpamcfg_mbw_winwd

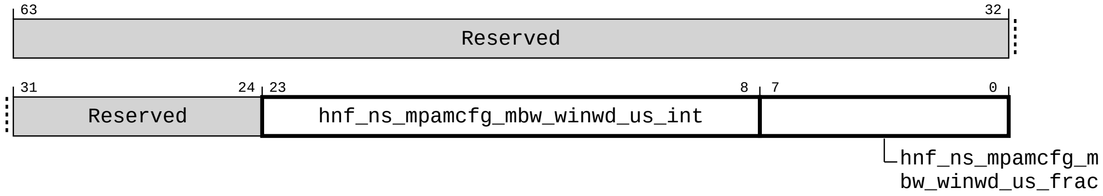

63
32
Reserved
31
24
23
8
7
0
Reserved
hnf_ns_mpamcfg_mbw_winwd_us_int
hnf_ns_mpamcfg_mbw_winwd_us_frac

Table 5-331: por_hnf_ns_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hnf_ns_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hnf_ns_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

5.3.12.22 por_hnf_ns_mpamcfg_pri

MPAM priority partitioning configuration register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-316: por_hnf_ns_mpamcfg_pri

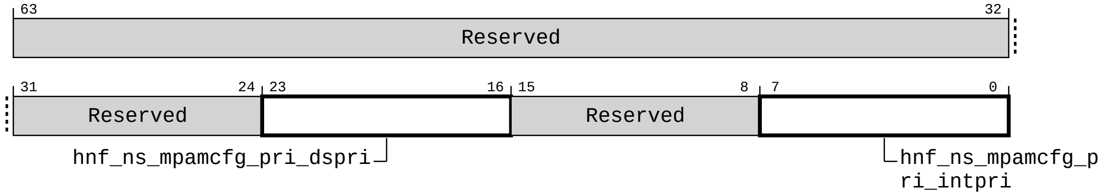


Table 5-332: por_hnf_ns_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hnf_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	-
[7:0]	hnf_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

5.3.12.23 por_hnf_ns_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-317: por_hnf_ns_mpamcfg_mbw_prop

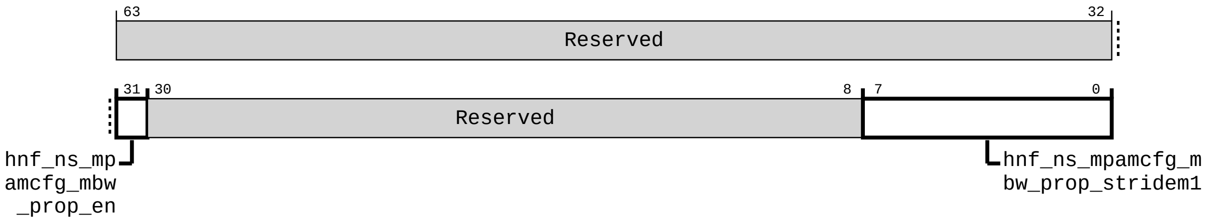


Table 5-333: por_hnf_ns_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_ns_mpamcfg_mbw_prop_stridem1	Normalized cost of bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	8'h0

5.3.12.24 por_hnf_ns_mpamcfg_intpartid

MPAM internal partition narrowing configuration register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-318: por_hnf_ns_mpamcfg_intpartid

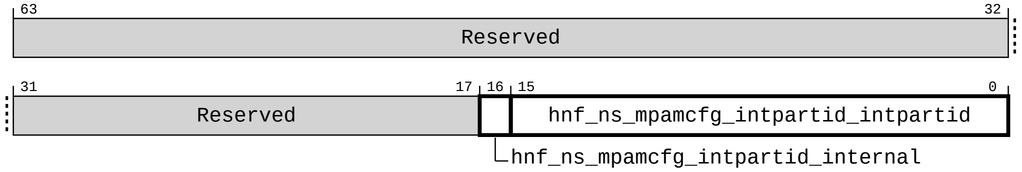

63
32
Reserved
31
17
16
15
0
Reserved
hnf_ns_mpamcfg_intpartid_intpartid
hnf_ns_mpamcfg_intpartid_internal

Table 5-334: por_hnf_ns_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hnf_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]	hnf_ns_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

5.3.12.25 por_hnf_ns_msmon_cfg_mon_sel

Memory system performance monitor selection register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-319: por_hnf_ns_msmon_cfg_mon_sel

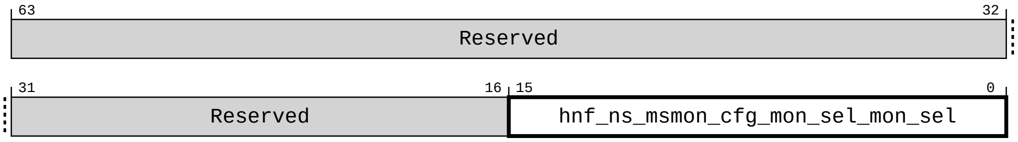


Table 5-335: por_hnf_ns_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_ns_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure	RW	16'h0

5.3.12.26 por_hnf_ns_msmon_capt_evnt

Memory system performance monitoring capture event generation register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-320: por_hnf_ns_msmon_capt_evnt

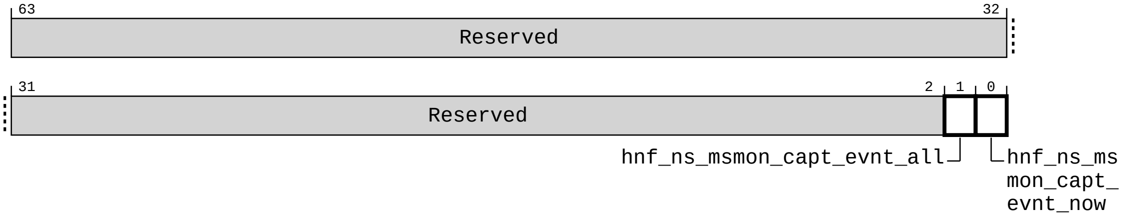


Table 5-336: por_hnf_ns_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	hnf_ns_msmon_capt_evnt_all	In Secure version, if ALL is written as 1 and NOW is also written as 1, it signals 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, it signals 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, it signals a capture event to Non-secure monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hnf_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

5.3.12.27 por_hnf_ns_msmon_cfg_csu_flt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-321: por_hnf_ns_msmon_cfg_csu_flt

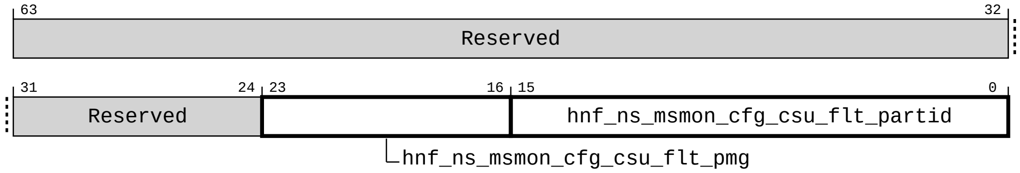


Table 5-337: por_hnf_ns_msmon_cfg_csu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hnf_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]	hnf_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

5.3.12.28 por_hnf_ns_msmon_cfg_csu_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-322: por_hnf_ns_msmon_cfg_csu_ctl

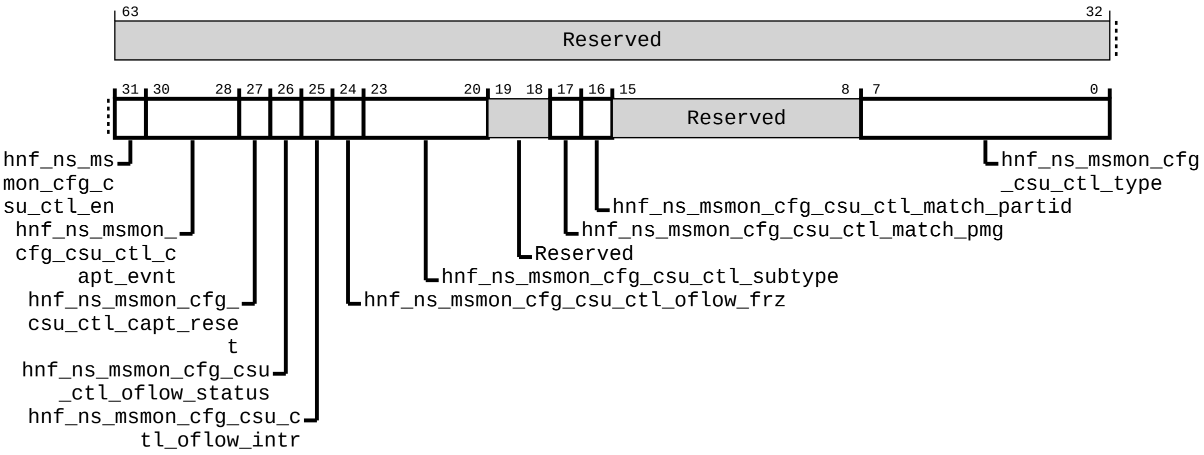

63
32
Reserved
31
30
28
27
26
25
24
23
20
19
18
17
16
15
Reserved
8
7
0
hnf_ns_msmon_cfg_csu_ctl_en
hnf_ns_msmon_cfg_csu_ctl_capt_evnt
hnf_ns_msmon_cfg_csu_ctl_capt_reset
hnf_ns_msmon_cfg_csu_ctl_oflow_status
hnf_ns_msmon_cfg_csu_ctl_oflow_intr
hnf_ns_msmon_cfg_csu_ctl_oflow_frz
hnf_ns_msmon_cfg_csu_ctl_subtype
Reserved
hnf_ns_msmon_cfg_csu_ctl_match_pmg
hnf_ns_msmon_cfg_csu_ctl_match_partid
hnf_ns_msmon_cfg_csu_ctl_type

Table 5-338: por_hnf_ns_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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]	hnf_ns_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	hnf_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]	hnf_ns_msmon_cfg_csu_ctl_oflow_intr	0: No interrupt 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hnf_ns_msmon_cfg_csu_ctl_oflow_frz	0: Monitor count wraps on overflow 1: Monitor count freezes on overflow. The frozen value can be 0 or another value if the monitor overflowed with an increment larger than 1.	RW	1'h0
[23:20]	hnf_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]	hnf_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

Bits	Name	Description	Type	Reset
[16]	hnf_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]	hnf_ns_msmon_cfg_csu_ctl_type	Read-only: Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	8'h43

5.3.12.29 por_hnf_ns_msmon_cfg_mbwu_flt

Memory system performance monitor configure memory bandwidth usage monitor filter register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-323: por_hnf_ns_msmon_cfg_mbwu_flt

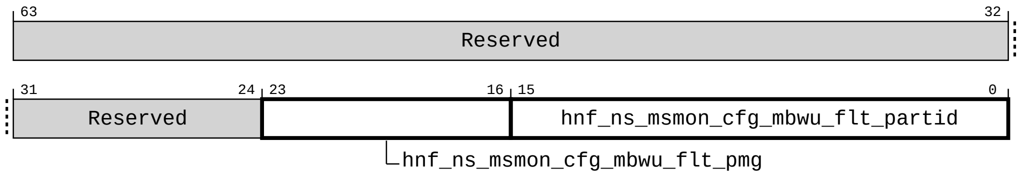

63
32
Reserved
31
24
23
16
15
0
Reserved
hnf_ns_msmon_cfg_mbwu_flt_partid
hnf_ns_msmon_cfg_mbwu_flt_pmg

Table 5-339: por_hnf_ns_msmon_cfg_mbwu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[23:16]	hnf_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]	hnf_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

5.3.12.30 por_hnf_ns_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-324: por_hnf_ns_msmon_cfg_mbwu_ctl

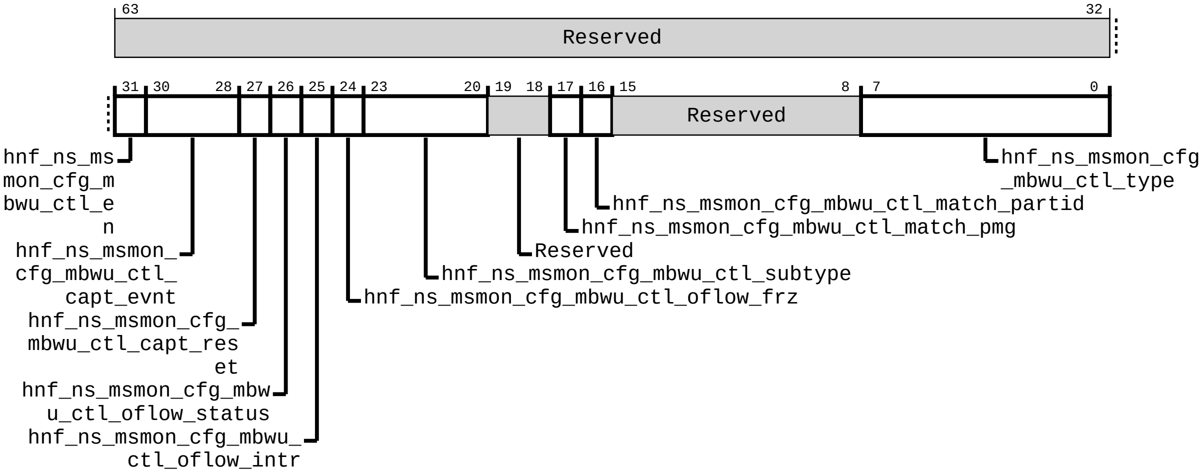

63
32
Reserved
31
30
28
27
26
25
24
23
20
19
18
17
16
15
Reserved
8
7
0
hnf_ns_msmon_cfg_mbwu_ctl_en
hnf_ns_msmon_cfg_mbwu_ctl_capt_evnt
hnf_ns_msmon_cfg_mbwu_ctl_capt_reset
hnf_ns_msmon_cfg_mbwu_ctl_oflow_status
hnf_ns_msmon_cfg_mbwu_ctl_oflow_intr
hnf_ns_msmon_cfg_mbwu_ctl_oflow_frz
hnf_ns_msmon_cfg_mbwu_ctl_subtype
Reserved
hnf_ns_msmon_cfg_mbwu_ctl_match_pmg
hnf_ns_msmon_cfg_mbwu_ctl_match_partid
hnf_ns_msmon_cfg_mbwu_ctl_type

Table 5-340: por_hnf_ns_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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]	hnf_ns_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hnf_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]	hnf_ns_msmon_cfg_mbwu_ctl_oflow_intr	0: No interrupt 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hnf_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]	hnf_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]	hnf_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]	hnf_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]	hnf_ns_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

5.3.12.31 por_hnf_ns_msmon_csu

Memory system performance monitor cache storage usage monitor register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-325: por_hnf_ns_msmon_csu

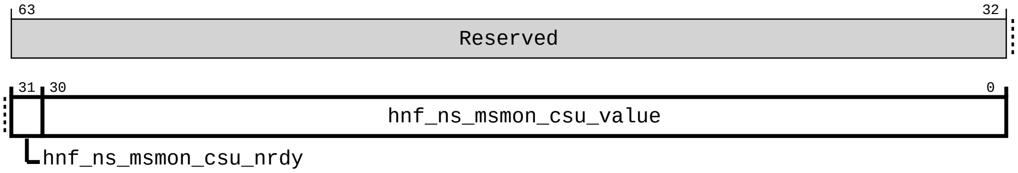


Table 5-341: por_hnf_ns_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.12.32 por_hnf_ns_msmon_csu_capture

Memory system performance monitor cache storage usage capture register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-326: por_hnf_ns_msmon_csu_capture

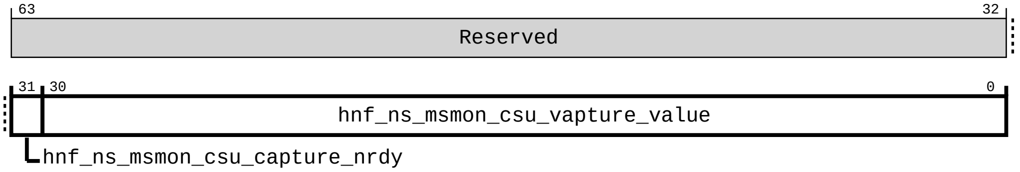


Table 5-342: por_hnf_ns_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.12.33 por_hnf_ns_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-327: por_hnf_ns_msmon_mbwu

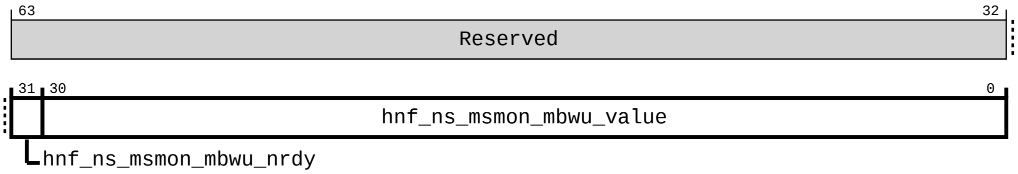


Table 5-343: por_hnf_ns_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.12.34 por_hnf_ns_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-328: por_hnf_ns_msmon_mbwu_capture

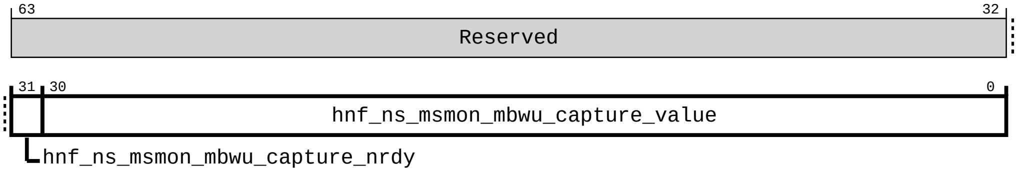

63
32
Reserved
31
30
0
hnf_ns_msmon_mbwu_capture_value
hnf_ns_msmon_mbwu_capture_nrdy

Table 5-344: por_hnf_ns_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hnf_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]	hnf_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

5.3.12.35 por_hnf_ns_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked seperately for S and NS.

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

This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-329: por_hnf_ns_mpamcfg_cpbm

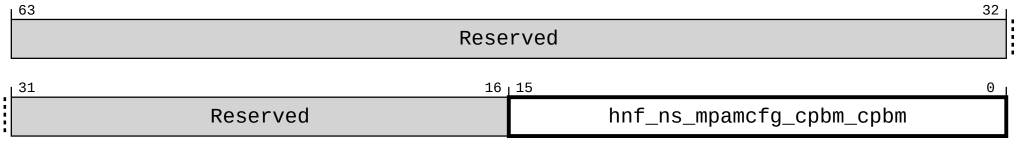


Table 5-345: por_hnf_ns_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_ns_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. Note: CPBM cannot be all zeros for any PARTID.	RW	16'hFFFF

5.3.13 RN-I register descriptions

This section lists the RN-I registers.

5.3.13.1 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

Figure 5-330: por_rni_node_info

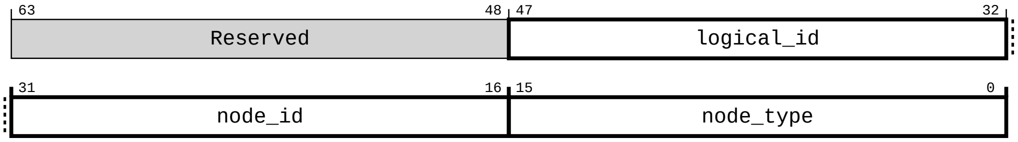


Table 5-346: por_rni_node_info attributes

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-650 node type identifier	RO	16'h000A

5.3.13.2 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

Figure 5-331: por_rni_child_info

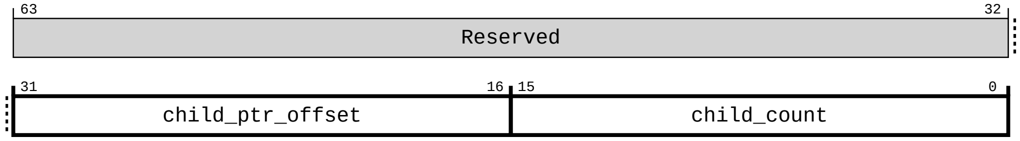


Table 5-347: por_rni_child_info attributes

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. This value is used in the discovery process.	RO	16'h0

5.3.13.3 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

Figure 5-332: por_rni_secure_register_groups_override

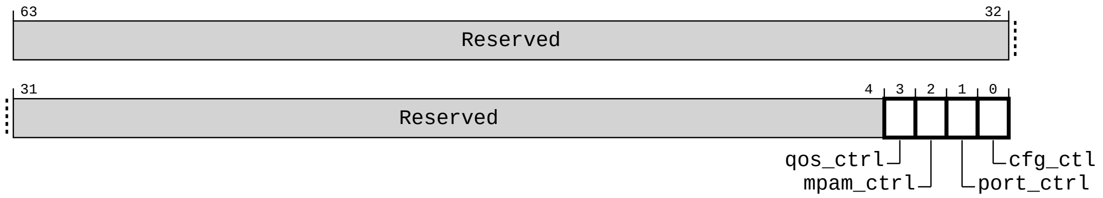


Table 5-348: por_rni_secure_register_groups_override attributes

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

5.3.13.4 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

Figure 5-333: por_rni_unit_info

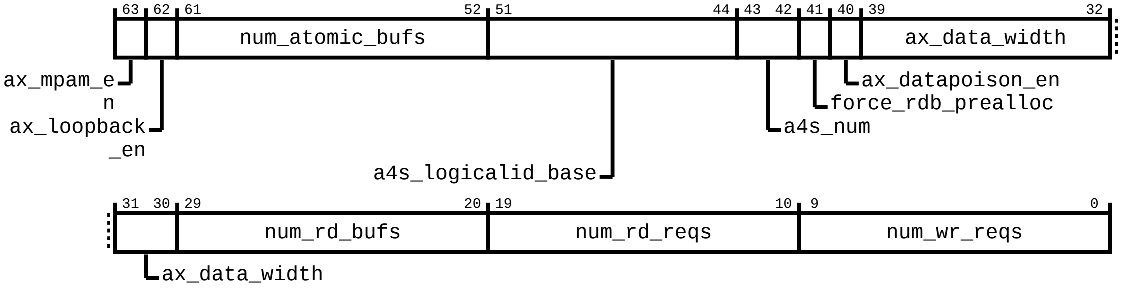


Table 5-349: por_rni_unit_info attributes

Bits	Name	Description	Type	Reset
[63]	ax_mpam_en	MPAM enable on AXI/ACE-Lite interface: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[62]	ax_loopback_en	LoopBack enable on AXI/ACE-Lite interface: 1'b1: Enabled 1'b0: Not enabled	RO	Configuration dependent
[61:52]	num_atomic_bufs	Number of atomic data buffers	RO	Configuration dependent
[51:44]	a4s_logicalid_base	AXI4-Stream interfaces logical ID base	RO	Configuration dependent
[43:42]	a4s_num	Number of AXI4-Stream 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 AXI/ACE-Lite 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

5.3.13.5 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

Figure 5-334: por_rni_unit_info2

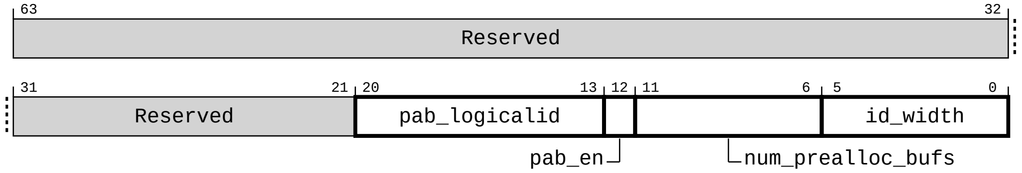


Table 5-350: por_rni_unit_info2 attributes

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[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 preallocated read data buffers	RO	Configuration dependent
[5:0]	id_width	AXI ID width for ACE-Lite slave ports	RO	Configuration dependent

5.3.13.6 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

Figure 5-335: por_rni_cfg_ctl

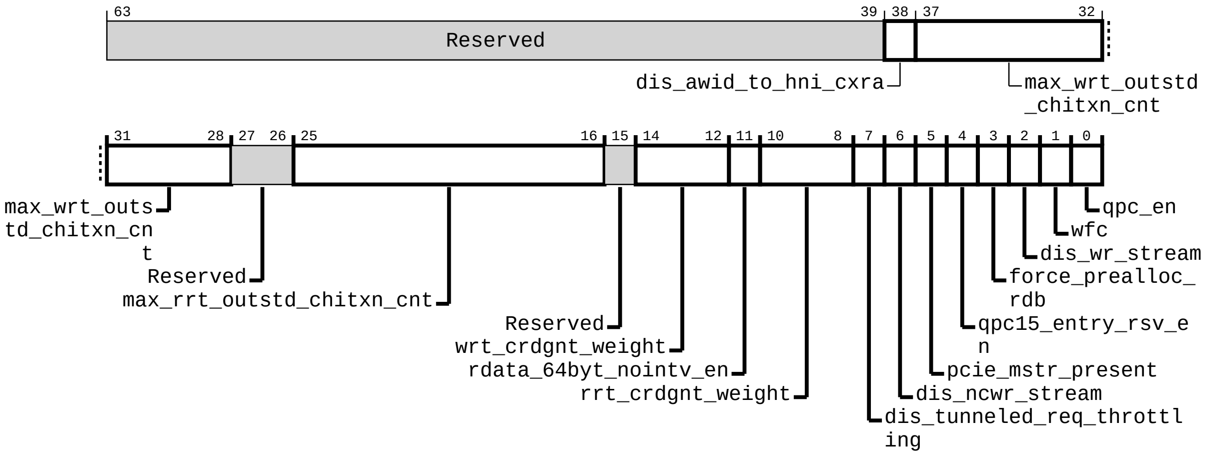


Table 5-351: por_rni_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:39]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[38]	dis_awid_to_hni_cxra	If set, disables compressed AWID to HN-I and CXRA, and disables compressed AWID-based ordering. Set this bit if unique-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 that is 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	RW	1'b1
[10:8]	rrt_crdgnt_weight	Determines weight of credit grant that is 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_ncwr_stream	If set, disables streaming of ordered non-cacheable writes	RW	1'b0
[5]	pcie_mstr_present	Indicates PCIe master is present. This bit must be set if PCIe master is present upstream of RN-I or RN-D.	RW	1'b0
[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	If set, all reads from the RN-I are sent with a preallocated read data buffer	RW	Configuration dependent
[2]	dis_wr_stream	If set, disables streaming of ordered writes	RW	1'b0
[1]	wfc	If set, enables waiting for completion (COMP) before dispatching dependent transaction (TXN)	RW	1'b0
[0]	qpc_en	If set, enables QoS Priority Class (QPC)-based scheduling using two QPCs (QoS15 and non-QoS15)	RW	1'b1

5.3.13.7 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

Figure 5-336: por_rni_aux_ctl

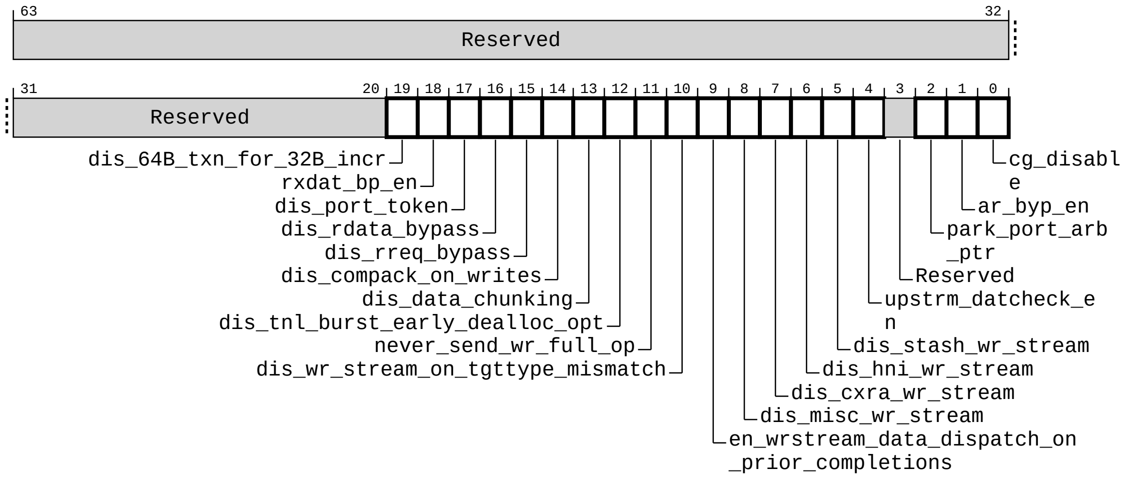


Table 5-352: por_rni_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:20]	Reserved	Reserved	RO	-
[19]	dis_64B_txn_for_32B_incr	If set, disables the 2nd, 32B fake write for a 32B INCR burst (rh-2512)	RW	1'b0
[18]	rxdat_bp_en	If set, backpressures the RXDAT interface when RDBs are not available	RW	1'b0
[17]	dis_port_token	If set, disables per port reservation in the tracker (read and write)	RW	1'b1
[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 transactions of a burst	RW	1'b0
[11]	never_send_wr_full_op	If set, RN-I never sends write FULL operations. All write operationss are of PTL type	RW	1'b0

Bits	Name	Description	Type	Reset
[10]	dis_wr_stream_on_tgttype_mismatch	If set, serializes first write when moving from one target type 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	If set, disables streaming of ordered writes with following attributes: Device memory or EWA=0 or Excl=1	RW	1'b0
[7]	dis_cxra_wr_stream	If set, disables streaming of ordered writes to CXRA	RW	1'b0
[6]	dis_hni_wr_stream	If set, disables streaming of ordered writes to HN-I	RW	1'b0
[5]	dis_stash_wr_stream	If set, disables streaming of ordered WrUniqStash	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	If set, disables clock gating	RW	1'b0

5.3.13.8 por_rni_s0-2_port_control

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls port S#{index} AXI/ACE slave interface settings.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA10 + (8 × #[0, 1, 2])

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

Figure 5-337: por_rni_s0-2_port_control

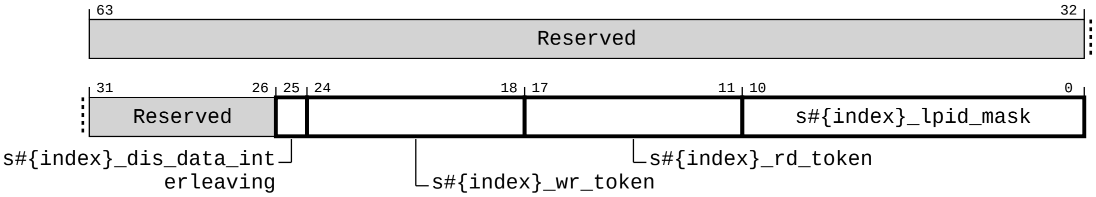


Table 5-353: por_rni_s0-2_port_control attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	s#{index}_dis_data_interleaving	If set, disables read data interleaving on RDATAS#{index} channel. This setting applies only to RDATA that is generated as a response to requests on the AR channel . This setting does not apply to RDATA that is generated as a response to atomic requests on the AW channel. In other words, RDATA of an atomic op on the AW channel can interleave with RDATA of an AR channel request.	RW	1'b0
[24:18]	s#{index}_wr_token	Port S#{index} reserved token count for AW channel. This must be less than the number of write requests (RNID_NUM_XRT_REQ) on AW channel.	RW	6'b00_0000
[17:11]	s#{index}_rd_token	Port S#{index} reserved token count for AR channel per slice. This value should be less than the number of read requests (RNID_NUM_XRT_SLICE_REQ) per slice on AR channel	RW	6'b00_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

5.3.13.9 por_rni_s0-2_mpam_control

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls port S#{index} AXI/ACE slave interface MPAM override values

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA28 + (8 × #[0, 1, 2])

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

Figure 5-338: por_rni_s0-2_mpam_control

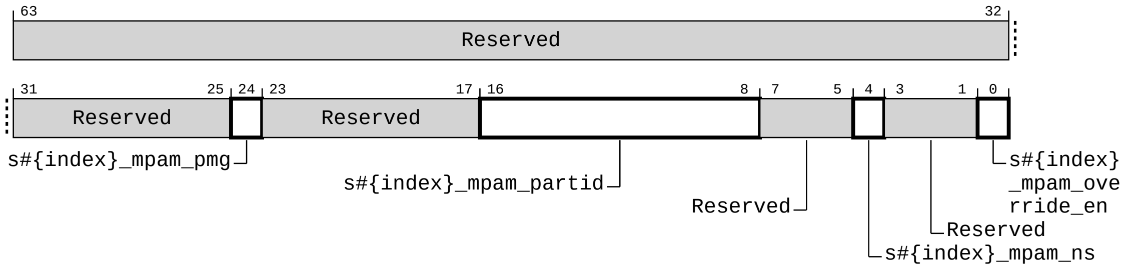


Table 5-354: por_rni_s0-2_mpam_control attributes

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. If 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, regardless of the value of this bit.	RW	1'b0

5.3.13.10 por_rni_s0-2_qos_control

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls QoS settings for port S#{index} AXI/ACE slave interface.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA80 + (32 × #[0, 1, 2])

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

Figure 5-339: por_rni_s0-2_qos_control

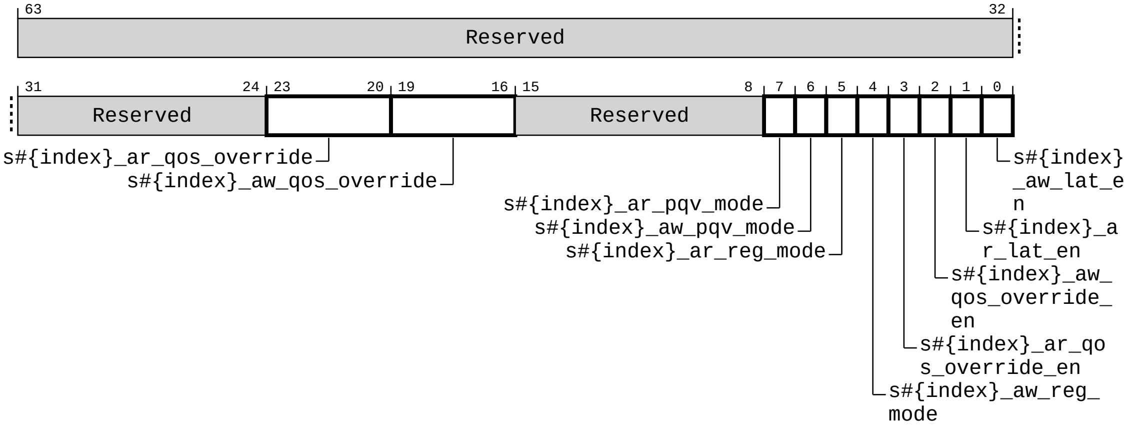


Table 5-355: por_rni_s0-2_qos_control attributes

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	-

Bits	Name	Description	Type	Reset
[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 master is idle. 1'b1: Quiesce high mode. QoS value tends to the maximum value when the master 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 master is idle. 1'b1: Quiesce high mode. QoS value tends to the maximum value when the master is idle.	RW	1'b0
[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. If 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. If set, allows QoS value on inbound AW transactions to be overridden.	RW	1'b0
[1]	s#{index}_ar_lat_en	If set, enables port S#{index} AR QoS regulation	RW	1'b0
[0]	s#{index}_aw_lat_en	If set, enables port S#{index} AW QoS regulation	RW	1'b0

5.3.13.11 por_rni_s0-2_qos_lat_tgt

There are 3 iterations of this register, parameterized by the index 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 + (32 × #[0, 1, 2])

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

Figure 5-340: por_rni_s0-2_qos_lat_tgt

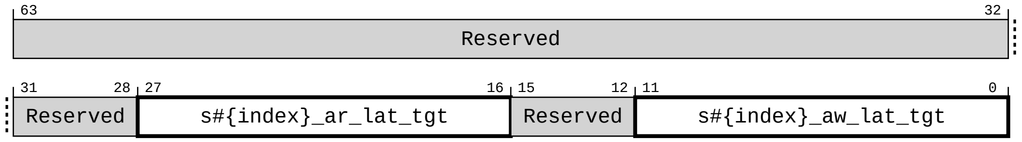


Table 5-356: por_rni_s0-2_qos_lat_tgt attributes

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

5.3.13.12 por_rni_s0-2_qos_lat_scale

There are 3 iterations of this register, parameterized by the index 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 + (32 × #[0, 1, 2])

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

Figure 5-341: por_rni_s0-2_qos_lat_scale

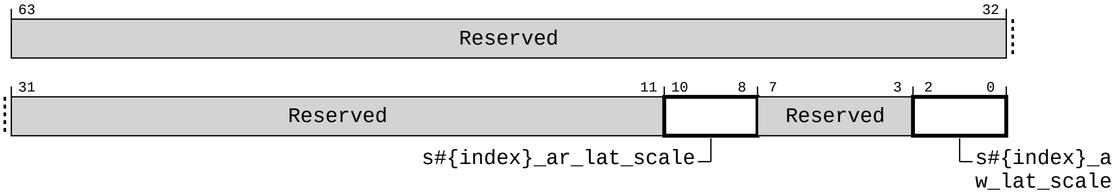


Table 5-357: por_rni_s0-2_qos_lat_scale attributes

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	-

Bits	Name	Description	Type	Reset
[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

5.3.13.13 por_rni_s0-2_qos_lat_range

There are 3 iterations of this register, parameterized by the index from 0 to 2. Controls the minimum and maximum QoS values that are 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 + (32 × #[0, 1, 2])

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

Figure 5-342: por_rni_s0-2_qos_lat_range

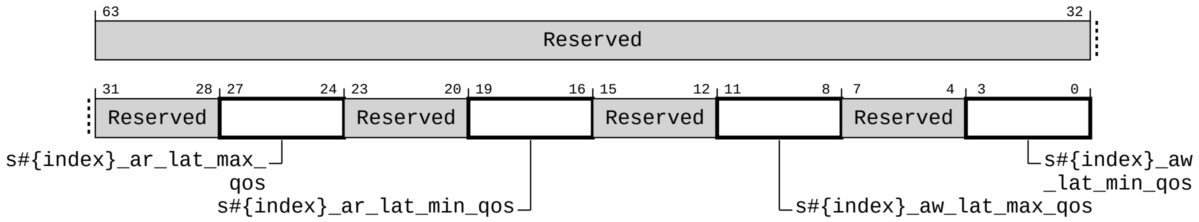


Table 5-358: por_rni_s0-2_qos_lat_range attributes

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

5.3.13.14 por_rni_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-343: por_rni_pmu_event_sel

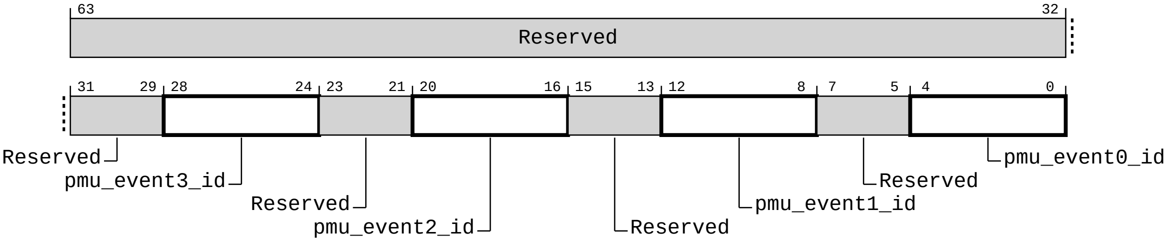


Table 5-359: por_rni_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:24]	pmu_event3_id	RN-I PMU event 3 ID. See pmu_event0_id for encodings.	RW	5'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	pmu_event2_id	RN-I PMU event 2 ID. See pmu_event0_id for encodings.	RW	5'b0
[15:13]	Reserved	Reserved	RO	-
[12:8]	pmu_event1_id	RN-I PMU event 1 ID. See pmu_event0_id for encodings.	RW	5'b0
[7:5]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[4:0]	pmu_event0_id	RN-I PMU event 0 ID 5'h00: No event 5'h01: Port S0 RDataBeats 5'h02: Port S1 RDataBeats 5'h03: Port S2 RDataBeats 5'h04: RXDAT flits received 5'h05: TXDAT flits sent 5'h06: Total TXREQ flits sent 5'h07: Retried TXREQ flits sent 5'h08: RRT occupancy count overflow 5'h09: WRT occupancy count overflow 5'h0A: Replayed TXREQ flits 5'h0B: WriteCancel sent 5'h0C: Port S0 WDataBeats 5'h0D: Port S1 WDataBeats 5'h0E: Port S2 WDataBeats 5'h0F: RRT allocation 5'h10: WRT allocation 5'h11: PADB occupancy count overflow 5'h12: RPDB occupancy count overflow 5'h13: RRT occupancy count overflow_slice1 5'h14: RRT occupancy count overflow_slice2 5'h15: RRT occupancy count overflow_slice3 5'h16: WRT request throttled	RW	5'b0

5.3.14 XP register descriptions

This section lists the XP registers.

5.3.14.1 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

Figure 5-344: por_mxp_node_info

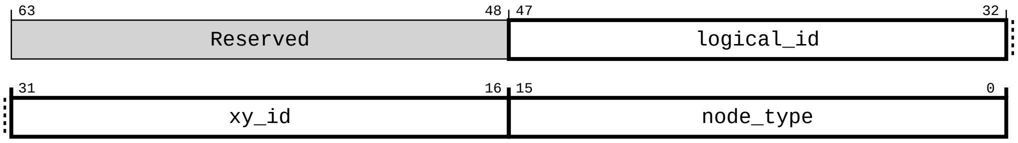


Table 5-360: por_mxp_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	Component logical ID	RO	Configuration dependent
[31:16]	xy_id	Identifies (X,Y) location of XP within the mesh. Note: The (X,Y) location is specified according to the node ID format as defined in Node ID mapping section of this TRM. 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 that represent the (X,Y) location varies for different node ID formats.	RO	16'h0000
[15:0]	node_type	CMN-650 node type identifier	RO	16'h0006

5.3.14.2 por_mxp_device_port_connect_info_p0-1

There are 2 iterations of this register, parameterized by the index from 0 to 1. Contains device port connection information for port #{index}

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h8 + (8 × #[0, 1])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-345: por_mxp_device_port_connect_info_p0-1

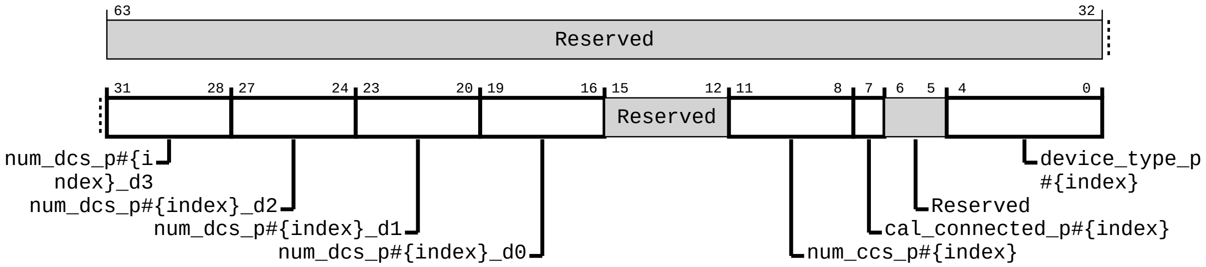


Table 5-361: por_mxp_device_port_connect_info_p0-1 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	num_dcs_p#{index}_d3	Number of Device Credited Slices (DCSs) connected to port #{index} device 3. Permitted values: 0-4.	RO	Configuration dependent
[27:24]	num_dcs_p#{index}_d2	Number of DCSs connected to port #{index} device 2. Permitted values: 0-4.	RO	Configuration dependent
[23:20]	num_dcs_p#{index}_d1	Number of DCSs connected to port #{index} device 1. Permitted values: 0-4.	RO	Configuration dependent
[19:16]	num_dcs_p#{index}_d0	Number of DCSs connected to port #{index} device 0. Permitted values: 0-4.	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[15:12]	Reserved	Reserved	RO	-
[11:8]	num_ccs_p#{index}	Number of CAL Credited Slices (CCSs) connected to port #{index}. Permitted values: 0-2.	RO	Configuration dependent
[7]	cal_connected_p#{index}	If set, CAL is connected on port #{index}. Permitted 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	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[4:0]	device_type_p#{index}	5'b11010: Reserved 5'b11011: Reserved 5'b11100-5'b11111: Reserved	RO	Configuration dependent

5.3.14.3 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'h18

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-346: por_mxp_mesh_port_connect_info_east

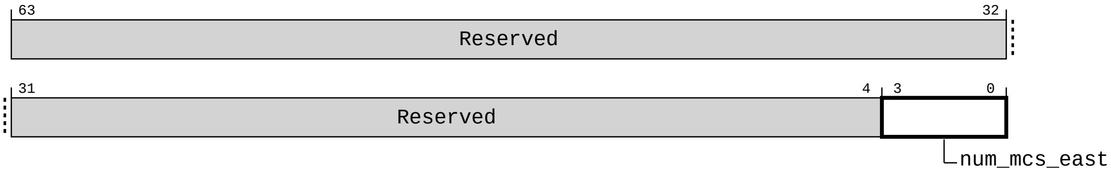


Table 5-362: por_mxp_mesh_port_connect_info_east attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	num_mcs_east	Number of Mesh Credited Slices (MCSs) connected to east port. Permitted values: 0-4.	RO	Configuration dependent

5.3.14.4 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'h20

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-347: por_mxp_mesh_port_connect_info_north

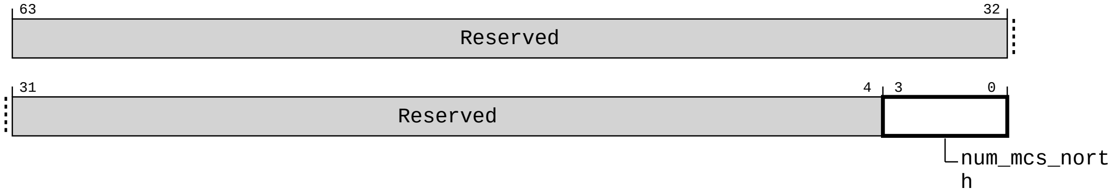


Table 5-363: por_mxp_mesh_port_connect_info_north attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	num_mcs_north	Number of Mesh Credited Slices (MCSs) connected to north port. Permitted values: 0-4.	RO	Configuration dependent

5.3.14.5 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

Figure 5-348: por_mxp_child_info

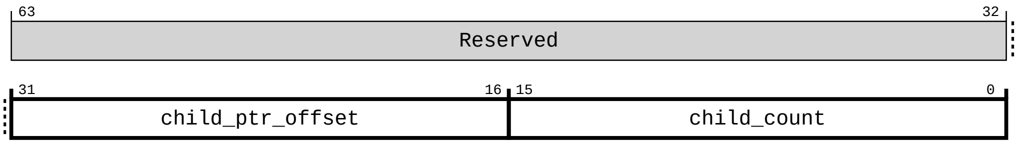


Table 5-364: por_mxp_child_info attributes

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. This value is used in the discovery process.	RO	Configuration dependent

5.3.14.6 por_mxp_child_pointer_0-31

There are 32 iterations of this register, parameterized by the index from 0 to 31. Contains base address of the configuration slave for child #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h100 + (8 × #[0, 1, ... 31])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-349: por_mxp_child_pointer_0-31

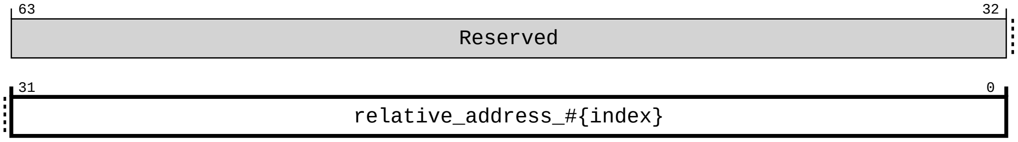


Table 5-365: por_mxp_child_pointer_0-31 attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	relative_address_#{index}	Bit [31]: External or internal child node. 1'b1: Indicates this child pointer points to a configuration node that is external to CMN-650 1'b0: Indicates this child pointer points to a configuration node that is internal to CMN-650 Bit [30]: Set to 1'b0 Bits [29:0]: Child node address offset relative to PERIPHBASE	RO	32'b0

5.3.14.7 por_mxp_p0-1_info

There are 2 iterations of this register, parameterized by the index from 0 to 1. Provides component identification information for XP port #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h900 + (8 × #[0, 1])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-350: por_mxp_p0-1_info

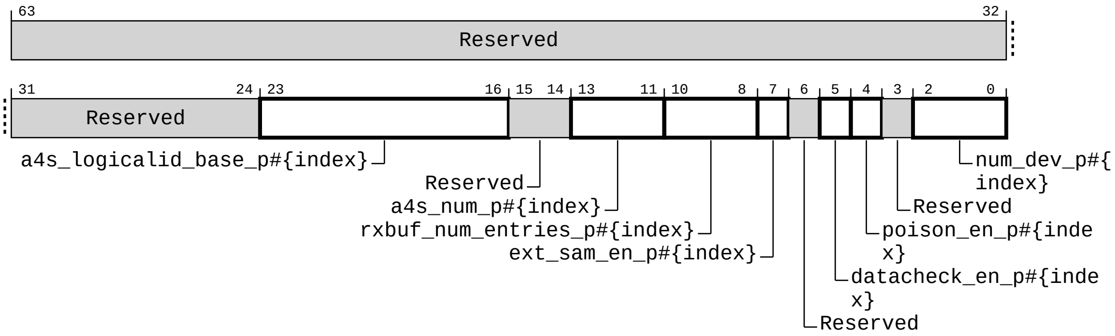


Table 5-366: por_mxp_p0-1_info attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	a4s_logicalid_base_p#{index}	AXI4-Stream interfaces logical ID base at this port (0 or 1)	RO	Configuration dependent
[15:14]	Reserved	Reserved	RO	-
[13:11]	a4s_num_p#{index}	Total number of RN-F AXI4-Stream interfaces at this port (0-4)	RO	Configuration dependent
[10:8]	rxbuf_num_entries_p#{index}	Number of input buffers at this port (2-4)	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 (0-4)	RO	Configuration dependent

5.3.14.8 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'h910

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-351: por_dtm_unit_info

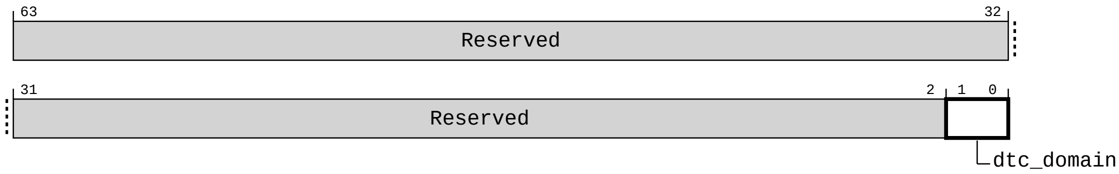


Table 5-367: por_dtm_unit_info attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	dtc_domain	DTC domain number associated with this DTM	RO	Configuration dependent

5.3.14.9 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

Figure 5-352: por_mxp_secure_register_groups_override

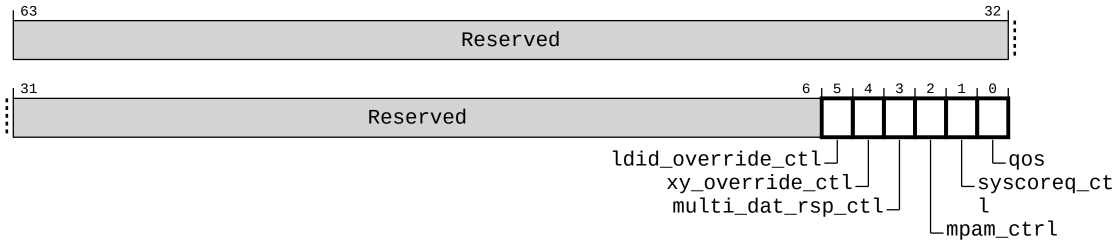


Table 5-368: por_mxp_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:6]	Reserved	Reserved	RO	-
[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_dat_rsp_ctl	Allows Non-secure access to Secure multi DAT and RSP control registers	RW	1'b0
[2]	mpam_ctrl	Allows Non-secure access to Secure CHI port MPAM override register	RW	1'b0
[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

5.3.14.10 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

Figure 5-353: por_mxp_aux_ctl

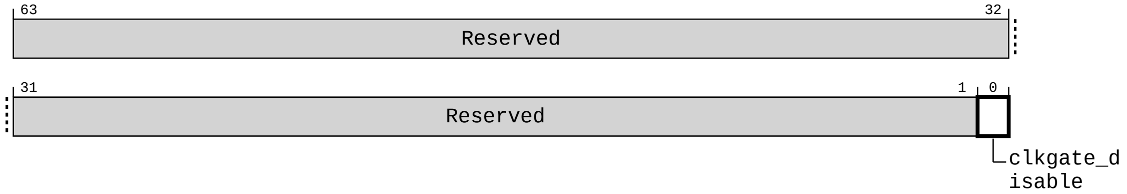


Table 5-369: por_mxp_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	clkgate_disable	Disables clock gating when set	RW	1'b0

5.3.14.11 por_mxp_p0-1_mpam_override

There are 2 iterations of this register, parameterized by the index from 0 to 1. Controls MPAM fields for devices connected to port #{index}. Valid only if the devices do not support MPAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA08 + (8 × #[0, 1])

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

Figure 5-354: por_mxp_p0-1_mpam_override

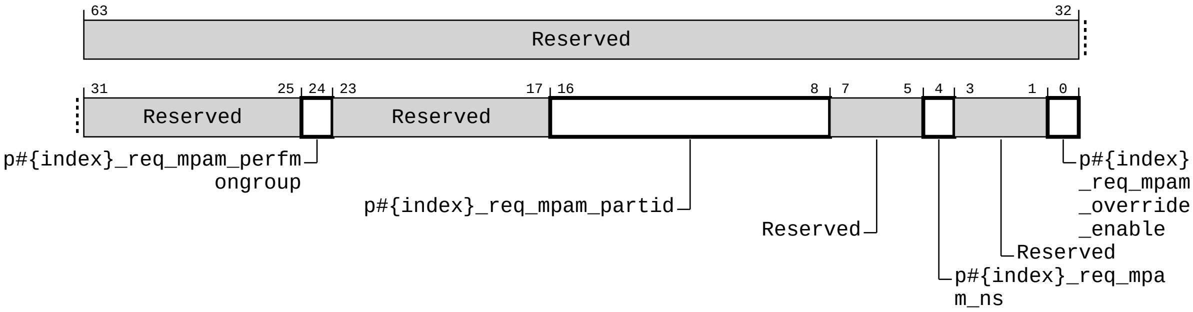

63
32
Reserved
31
25
24
23
17
16
8
7
5
4
3
1
0
Reserved
Reserved
p#{index}_req_mpam_perfmongroup
p#{index}_req_mpam_partid
Reserved
p#{index}_req_mpam_override_enable
Reserved
p#{index}_req_mpam_ns

Table 5-370: por_mxp_p0-1_mpam_override attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	p#{index}_req_mpam_perfmongroup	MPAM.PerfMonGroup subfield 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 subfield 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 subfield 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

5.3.14.12 por_mxp_p0-1_ldid_override

There are 2 iterations of this register, parameterized by the index from 0 to 1. 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'hA38 + (8 × #[0, 1])

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

Figure 5-355: por_mxp_p0-1_ldid_override

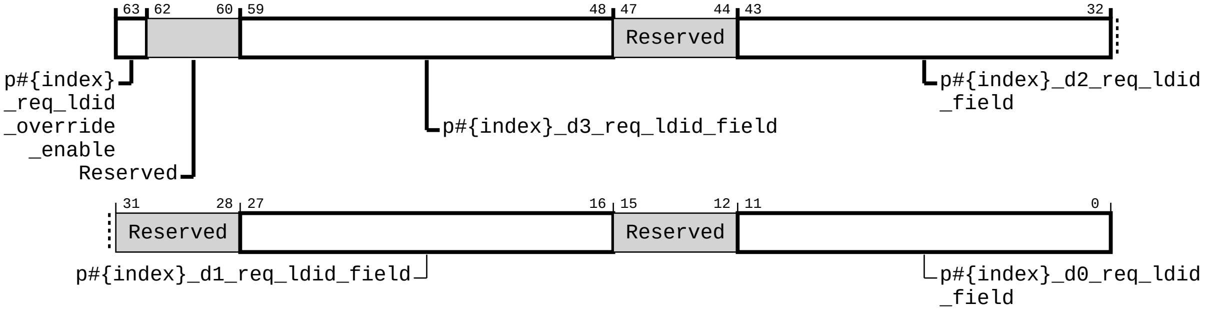

63
62
60
59
48
47
44
43
32
Reserved
p#{index}_req_ldid_override_enable
Reserved
p#{index}_d3_req_ldid_field
p#{index}_d2_req_ldid_field
31
28
27
16
15
12
11
0
Reserved
Reserved
p#{index}_d1_req_ldid_field
p#{index}_d0_req_ldid_field

Table 5-371: por_mxp_p0-1_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	Configuration dependent
[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	Configuration dependent
[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	Configuration dependent

Bits	Name	Description	Type	Reset
[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	Configuration dependent

5.3.14.13 por_mxp_p0-1_qos_control

There are 2 iterations of this register, parameterized by the index from 0 to 1. 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 × #[0, 1])

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

Figure 5-356: por_mxp_p0-1_qos_control

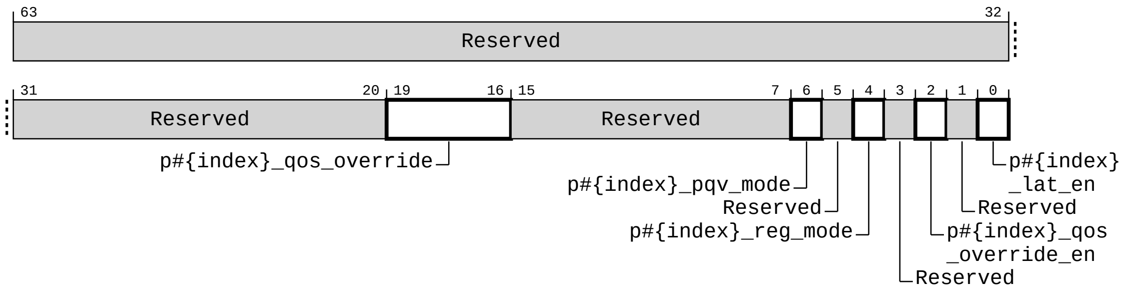


Table 5-372: por_mxp_p0-1_qos_control attributes

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 master is idle. 1'b1: Quiesce high mode. QoS value tends to the maximum value when the master 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. If 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

5.3.14.14 por_mxp_p0-1_qos_lat_tgt

There are 2 iterations of this register, parameterized by the index from 0 to 1. Controls QoS target latency and 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 × #[0, 1])

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

Figure 5-357: por_mxp_p0-1_qos_lat_tgt

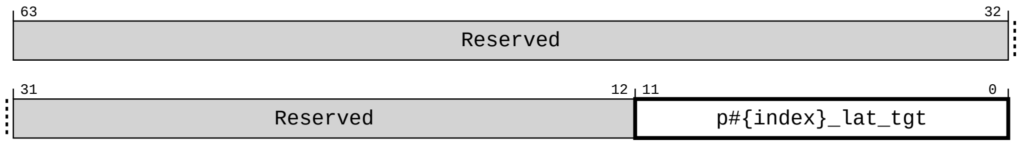


Table 5-373: por_mxp_p0-1_qos_lat_tgt attributes

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	-
[11:0]	p#{index}_lat_tgt	Port #{index} transaction target latency and period. A value of 0 corresponds to no regulation.	RW	12'h000

5.3.14.15 por_mxp_p0-1_qos_lat_scale

There are 2 iterations of this register, parameterized by the index from 0 to 1. 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 × #[0, 1])

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

Figure 5-358: por_mxp_p0-1_qos_lat_scale

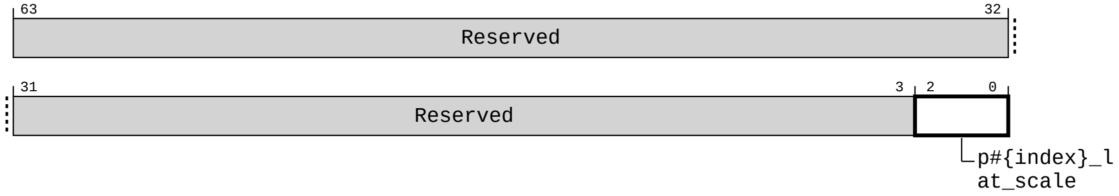


Table 5-374: por_mxp_p0-1_qos_lat_scale attributes

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

5.3.14.16 por_mxp_p0-1_qos_lat_range

There are 2 iterations of this register, parameterized by the index from 0 to 1. 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 × #[0, 1])

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

Figure 5-359: por_mxp_p0-1_qos_lat_range

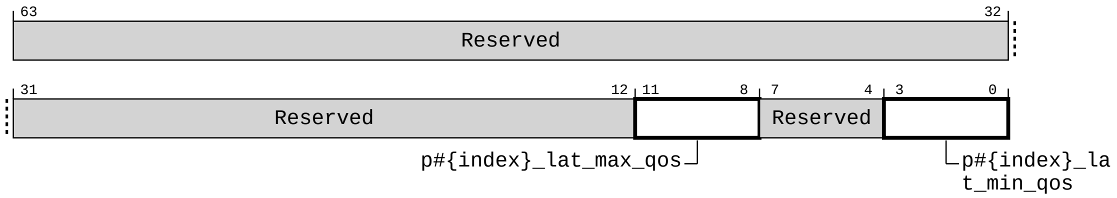


Table 5-375: por_mxp_p0-1_qos_lat_range attributes

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

5.3.14.17 por_mxp_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Figure 5-360: por_mxp_pmu_event_sel

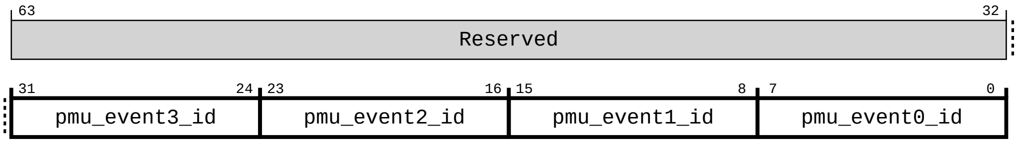


Table 5-376: por_mxp_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	pmu_event3_id	XP PMU event 3 ID. See pmu_event0_id for encodings.	RW	8'b0
[23:16]	pmu_event2_id	XP PMU event 2 ID. See pmu_event0_id for encodings.	RW	8'b0
[15:8]	pmu_event1_id	XP PMU event 1 ID. See pmu_event0_id for encodings.	RW	8'b0

Bits	Name	Description	Type	Reset
[7:0]	pmu_event0_id	XP PMU event 0 ID: Bits [7:5]: PC: 3'b000: REQ 3'b001: RSP; RSP channel when POR_RSP_VC_NUM_PARAM = 1 ; RSP subchannel 1: when POR_RSP_VC_NUM_PARAM > 1 3'b010: SNP 3'b011: DAT; DAT channel when POR_DAT_VC_NUM_PARAM = 1 ; DAT subchannel 1: when POR_DAT_VC_NUM_PARAM > 1 3'b100: PUB 3'b101: RSP2; RSP subchannel 2: Applicable when POR_RSP_VC_NUM_PARAM > 1 3'b110: DAT2; DAT subchannel 2: Applicable when POR_DAT_VC_NUM_PARAM > 1 Bits [4:2]: Interface: 3'b000: East when NUM_XP > 1 ; Device port 0 when NUM_XP == 1, in other words a single-MXP configuration 3'b001: West when NUM_XP > 1 ; Device port 1 when NUM_XP == 1, in other words a single-MXP configuration 3'b010: North when NUM_XP > 1 ; Device port 2 when NUM_XP == 1, in other words a single-MXP configuration 3'b011: South when NUM_XP > 1 ; Device port 3 when NUM_XP == 1, in other words a single-MXP configuration 3'b100: Device port 0 when NUM_XP > 1 ; Device port 4 when NUM_XP == 1, in other words a single-MXP configuration 3'b101: Device port 1 when NUM_XP > 1 ; Device port 5 when NUM_XP == 1, in other words a single-MXP configuration 3'b110: Device port 2 when NUM_XP > 1 ; No Selection when NUM_XP == 1, in other words a single-MXP configuration 3'b111: Device port 3 when NUM_XP > 1 ; No Selection when NUM_XP == 1, in other words a single-MXP configuration 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 CHI-A and RN-F CHI-A ESAM ports	RW	8'b0

5.3.14.18 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-361: por_mxp_errfr

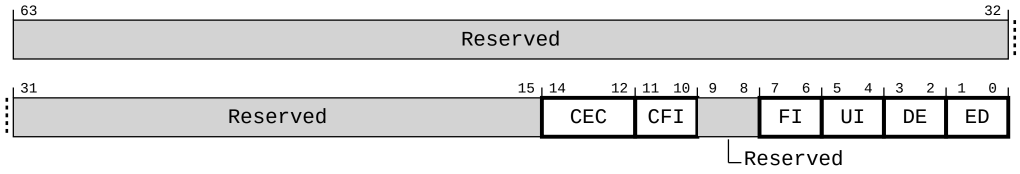


Table 5-377: por_mxp_errfr attributes

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

5.3.14.19 por_mxp_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-362: por_mxp_errctlr

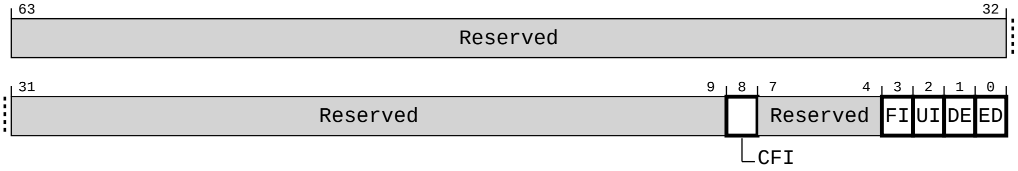


Table 5-378: por_mxp_errctlr attributes

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

5.3.14.20 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-363: por_mxp_errstatus

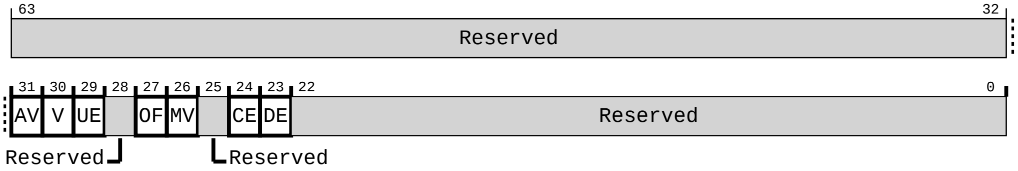


Table 5-379: por_mxp_errstatus attributes

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
[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

Bits	Name	Description	Type	Reset
[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 slave 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, and 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	-

5.3.14.21 por_mxp_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-364: por_mxp_errmisc

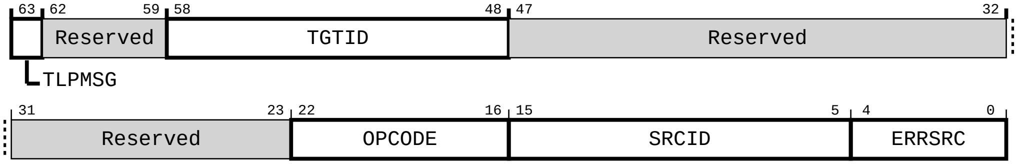


Table 5-380: por_mxp_errmisc attributes

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
[4:0]	ERRSRC	Error source: Bits [4:3]: Transaction type: 2'b00: REQ 2'b01: RSP 2'b10: SNP 2'b11: DAT Bits [2:0]: Port: 3'b000: Port 0 3'b001: Port 1 3'b010: Port 2 3'b011: Port 3 3'b100: Port 4 3'b101: Port 5	RW	5'b0

5.3.14.22 por_mxp_p0-1_byte_par_err_inj

There are 2 iterations of this register, parameterized by the index from 0 to 1. 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 × #[0, 1])

Type

WO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-365: por_mxp_p0-1_byte_par_err_inj

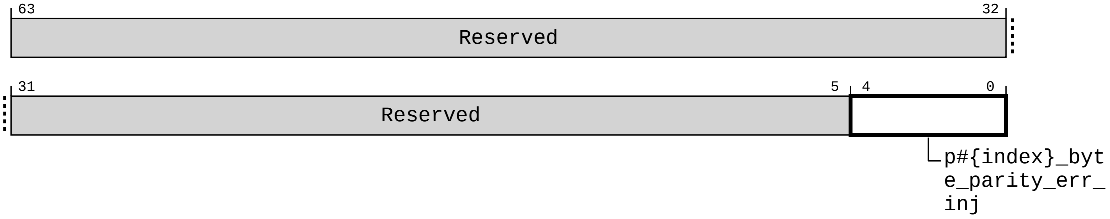


Table 5-381: por_mxp_p0-1_byte_par_err_inj attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:0]	p#{index}_byte_parity_err_inj	Specifies a byte lane. When 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

5.3.14.23 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

Figure 5-366: por_mxp_errfr_NS

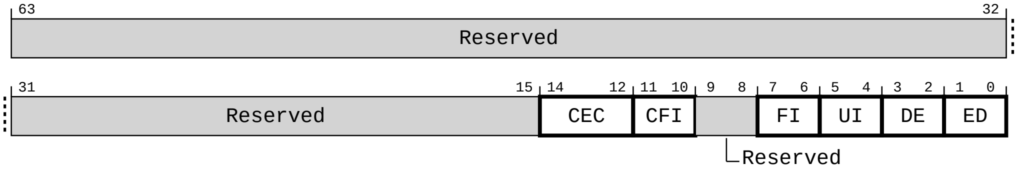


Table 5-382: por_mxp_errfr_NS attributes

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

5.3.14.24 por_mxp_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection and 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

Figure 5-367: por_mxp_errctlr_NS

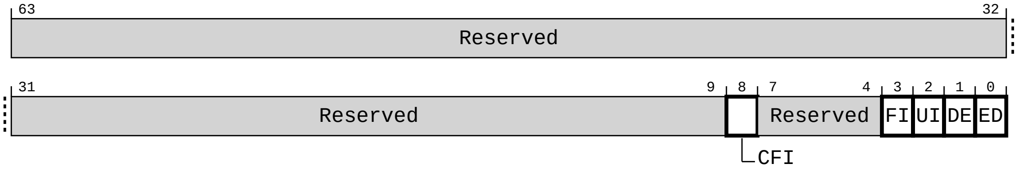


Table 5-383: por_mxp_errctlr_NS attributes

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

5.3.14.25 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

Figure 5-368: por_mxp_errstatus_NS

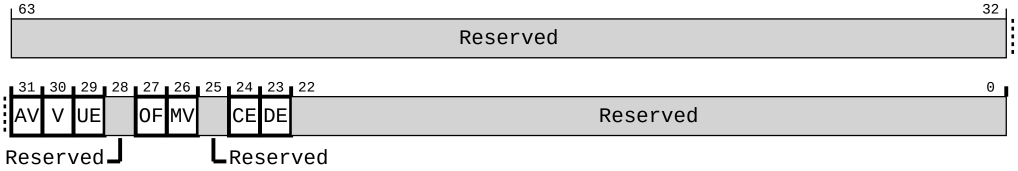


Table 5-384: por_mxp_errstatus_NS attributes

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

Bits	Name	Description	Type	Reset
[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 slave 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, and CE fields	W1C	1'b0
[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	-

5.3.14.26 por_mxp_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred and 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

Figure 5-369: por_mxp_errmisc_NS

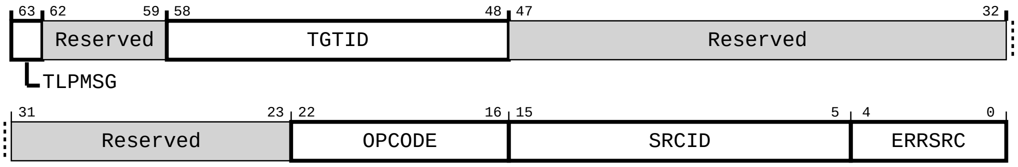


Table 5-385: por_mxp_errmisc_NS attributes

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. Bits [4:3]: Transaction type: 2'b00: REQ 2'b01: RSP 2'b10: SNP 2'b11: DAT Bits [2:0]: Port: 3'b000: Port 0 3'b001: Port 1 3'b010: Port 2 3'b011: Port 3 3'b100: Port 4 3'b101: Port 5	RW	5'b0

5.3.14.27 por_mxp_p0-1_syscoreq_ctl

There are 2 iterations of this register, parameterized by the index from 0 to 1. Functions as the port #{index} snoop and DVM domain control register.

Provides a software alternative to hardware SYSCOREQ, SYSCOACK handshake. Works with the por_mxp_p#{index}_syscoack_status register.

Note

Only valid on RN-F ports.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C00 + (8 × #[0, 1])

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

Figure 5-370: por_mxp_p0-1_syscoreq_ctl

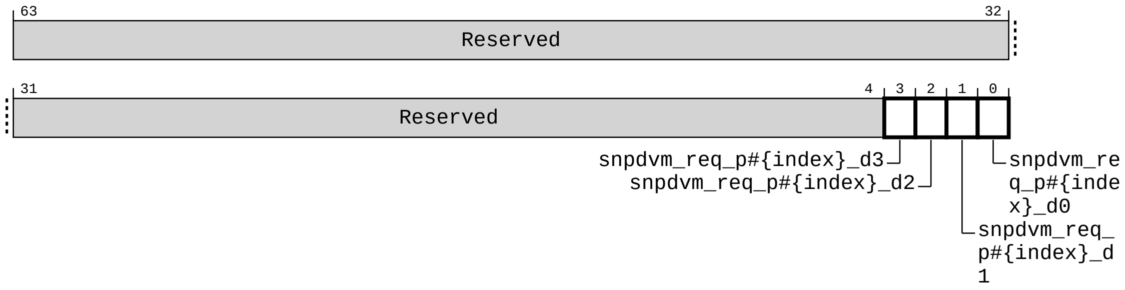


Table 5-386: por_mxp_p0-1_syscoreq_ctl attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	snpdvm_req_p#{index}_d3	If 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	If 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	If 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	If set, initiates the process of enabling snoop and DVM dispatches (SYSCOREQ) to device 0 on port #{index}	RW	1'b0

5.3.14.28 por_mxp_p0-1_syscoack_status

There are 2 iterations of this register, parameterized by the index from 0 to 1. 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'h1C10 + (8 × #[0, 1])

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

Figure 5-371: por_mxp_p0-1_syscoack_status

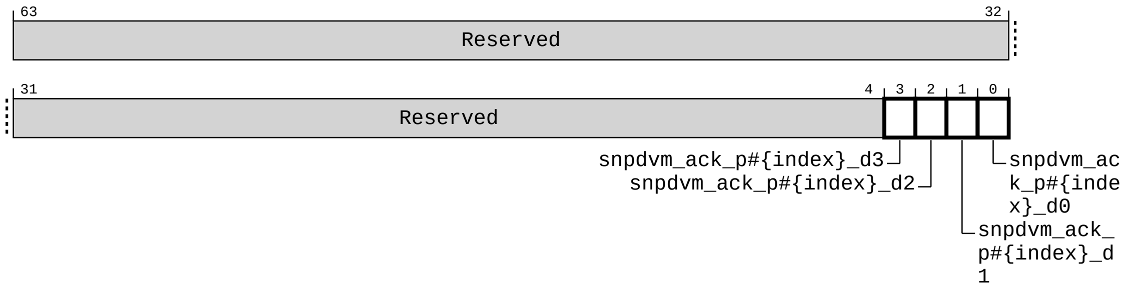


Table 5-387: por_mxp_p0-1_syscoack_status attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	snpdvm_ack_p#{index}_d3	If set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 3 on port #{index}	RO	1'b0
[2]	snpdvm_ack_p#{index}_d2	If set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 2 on port #{index}	RO	1'b0
[1]	snpdvm_ack_p#{index}_d1	If set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 1 on port #{index}	RO	1'b0
[0]	snpdvm_ack_p#{index}_d0	If set, indicates snoop and DVM dispatches are enabled (SYSCOACK) for device 0 on port #{index}	RO	1'b0

5.3.14.29 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

Figure 5-372: por_dtm_control

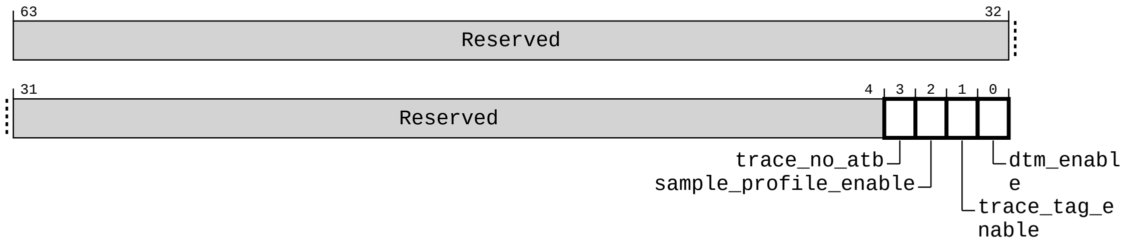


Table 5-388: por_dtm_control attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	trace_no_atb	If 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 is not 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. Before writing this bit, all other DT configuration registers must be programmed. After this bit is set, other DT configuration registers must not be modified.	RW	1'b0

5.3.14.30 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

Figure 5-373: por_dtm_fifo_entry_ready

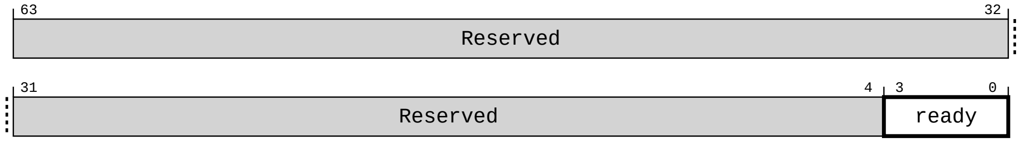


Table 5-389: por_dtm_fifo_entry_ready attributes

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]: If set, entry 3 ready Bit [2]: If set, entry 2 ready Bit [1]: If set, entry 1 ready Bit [0]: If set, entry 0 ready	W1C	4'b0

5.3.14.31 por_dtm_fifo_entry0-3_0

There are 4 iterations of this register, parameterized by the index 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 × #[0, 1, ... 3])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-374: por_dtm_fifo_entry0-3_0

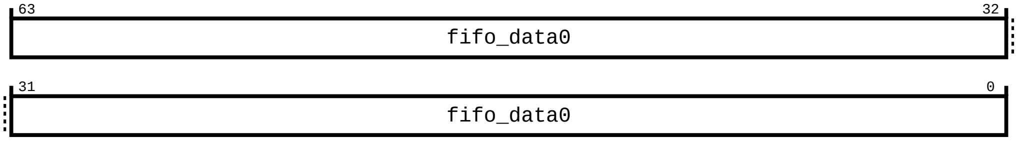


Table 5-390: por_dtm_fifo_entry0-3_0 attributes

Bits	Name	Description	Type	Reset
[63:0]	fifo_data0	Entry data bit vector[63:0]	RO	64'b0

5.3.14.32 por_dtm_fifo_entry0-3_1

There are 4 iterations of this register, parameterized by the index 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 × #[0, 1, ... 3])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-375: por_dtm_fifo_entry0-3_1

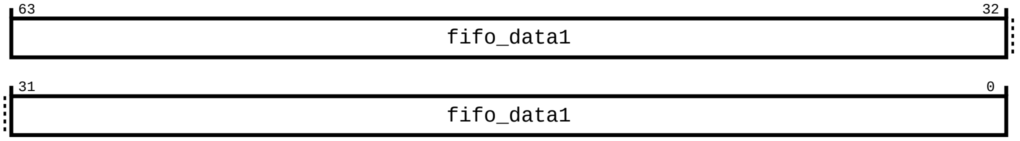


Table 5-391: por_dtm_fifo_entry0-3_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	fifo_data1	Entry data bit vector[127:64]	RO	64'b0

5.3.14.33 por_dtm_fifo_entry0-3_2

There are 4 iterations of this register, parameterized by the index 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 × #[0, 1, ... 3])

Type

RO

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-376: por_dtm_fifo_entry0-3_2

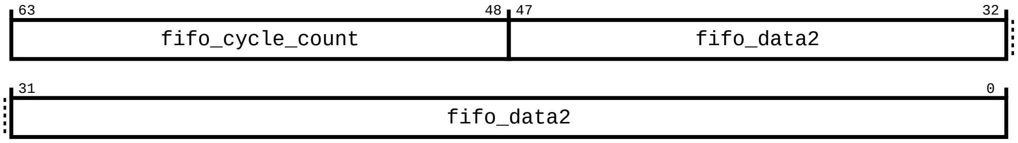


Table 5-392: por_dtm_fifo_entry0-3_2 attributes

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

5.3.14.34 por_dtm_wp0-3_config

There are 4 iterations of this register, parameterized by the index from 0 to 3. Configures watchpoint #{index}.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h21A0 + (24 × #[0, 1, ... 3])

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-377: por_dtm_wp0-3_config

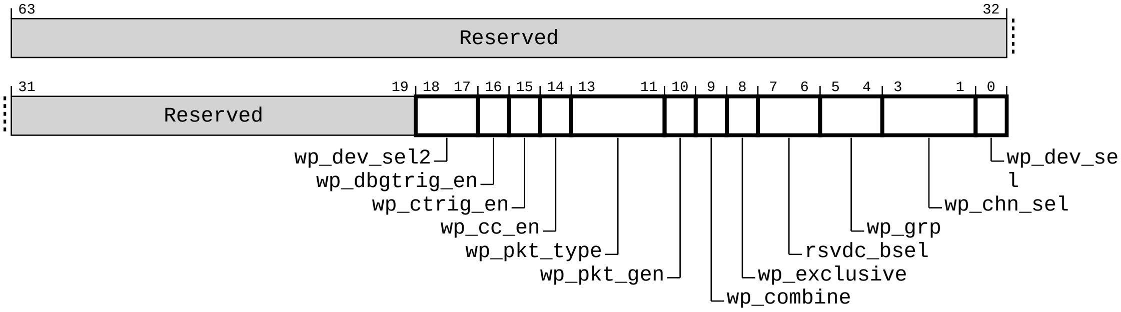


Table 5-393: por_dtm_wp0-3_config attributes

Bits	Name	Description	Type	Reset
[63:19]	Reserved	Reserved	RO	-
[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

Bits	Name	Description	Type	Reset
[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

Bits	Name	Description	Type	Reset
[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

5.3.14.35 por_dtm_wp0-3_val

There are 4 iterations of this register, parameterized by the index 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 × #[0, 1, ... 3])

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-378: por_dtm_wp0-3_val

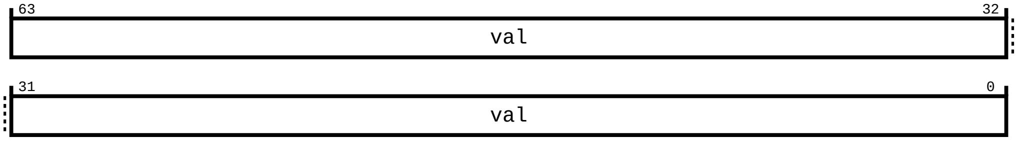


Table 5-394: por_dtm_wp0-3_val attributes

Bits	Name	Description	Type	Reset
[63:0]	val	See DTM watchpoint section for more information	RW	64'b0

5.3.14.36 por_dtm_wp0-3_mask

There are 4 iterations of this register, parameterized by the index 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 × #[0, 1, ... 3])

Type

RW

Reset value

See individual bit resets

Usage constraints

There are no usage constraints.

Bit descriptions

Figure 5-379: por_dtm_wp0-3_mask

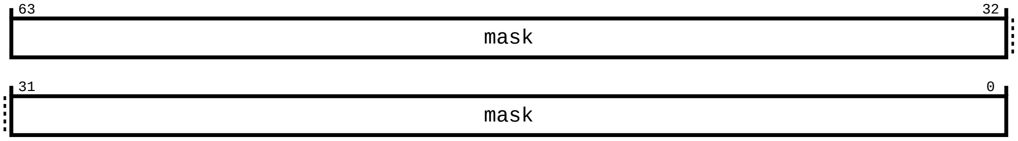


Table 5-395: por_dtm_wp0-3_mask attributes

Bits	Name	Description	Type	Reset
[63:0]	mask	See to DTM watchpoint section for more information	RW	64'b0

5.3.14.37 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

Figure 5-380: por_dtm_pmsicr

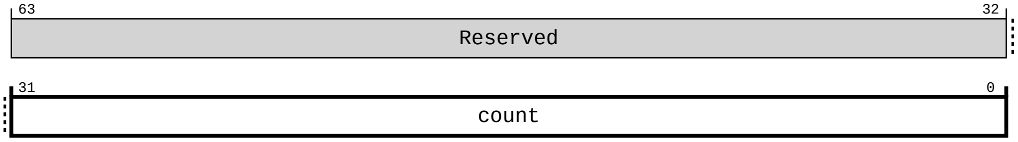


Table 5-396: por_dtm_pmsicr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	count	Current value of sample counter	RW	32'b0

5.3.14.38 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

Figure 5-381: por_dtm_pmsirr

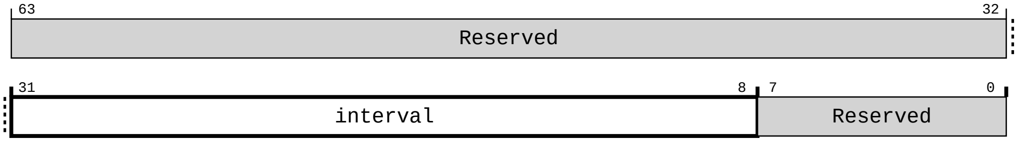


Table 5-397: por_dtm_pmsirr attributes

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	-

5.3.14.39 por_dtm_pmu_config

Configures the DTM Performance Monitoring Unit (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

Figure 5-382: por_dtm_pmu_config

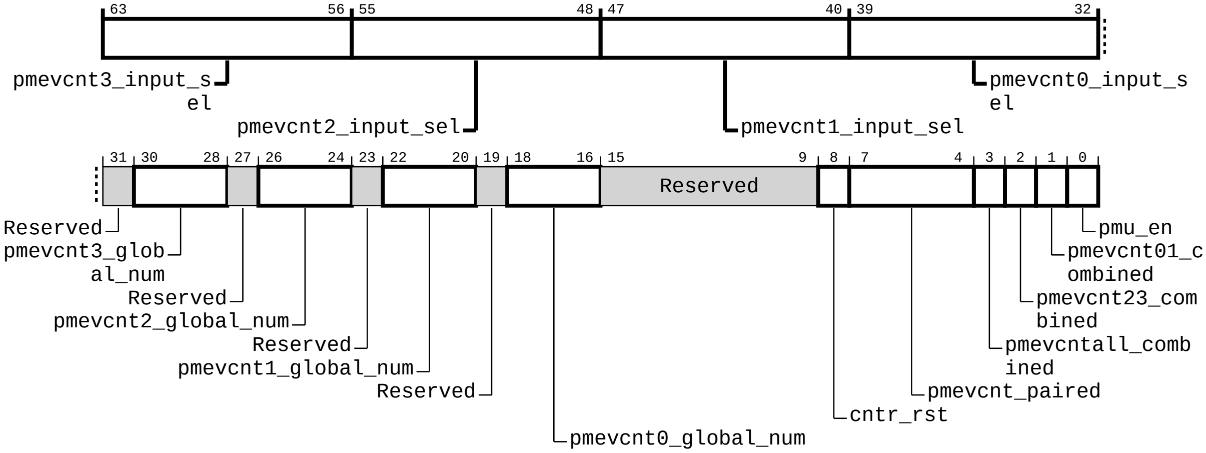


Table 5-398: por_dtm_pmu_config attributes

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

Bits	Name	Description	Type	Reset
[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 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	RW	8'b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	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 8'h30: Port 2 Device 0 PMU event 0 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	RW	8'b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	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 8'h40: Port 3 Device 0 PMU event 0 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 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	RW	8'b0

Bits	Name	Description	Type	Reset
[39:32]	pmevcnt0_input_sel	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 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

Bits	Name	Description	Type	Reset
[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
[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, and 3. NOTE: If 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

5.3.14.40 por_dtm_pmevcnt

Contains all PMU event counters, 0, 1, 2, and 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

Figure 5-383: por_dtm_pmevcnt

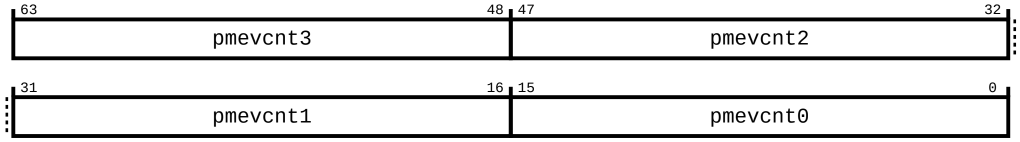


Table 5-399: por_dtm_pmevcnt attributes

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
[31:16]	pmevcnt1	PMU event counter 1	RW	16'h0000
[15:0]	pmevcnt0	PMU event counter 0	RW	16'h0000

5.3.14.41 por_dtm_pmevcntsr

Functions as the PMU event counter shadow register for all counters, 0, 1, 2, and 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

Figure 5-384: por_dtm_pmevcntsr

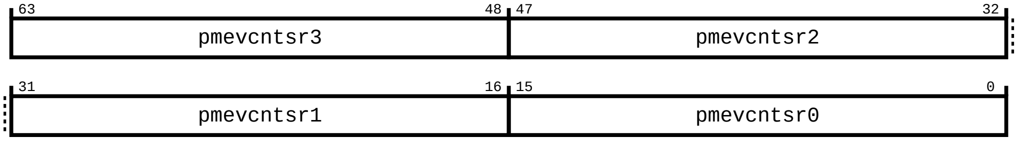


Table 5-400: por_dtm_pmevcntsr attributes

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

5.3.14.42 por_mxp_multi_dat_rsp_chn_sel_0-15

There are 16 iterations of this register, parameterized by the index from 0 to 15. Functions as the DAT and RSP channel select register for the dual DAT/RSP feature per target for the XP.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC00 + (8 × #[0, 1, ... 15])

Type

RW

Reset value

See individual bit resets

Secure group override

por_mxp_secure_register_groups_override.multi_dat_rsp_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

Figure 5-385: por_mxp_multi_dat_rsp_chn_sel_0-15

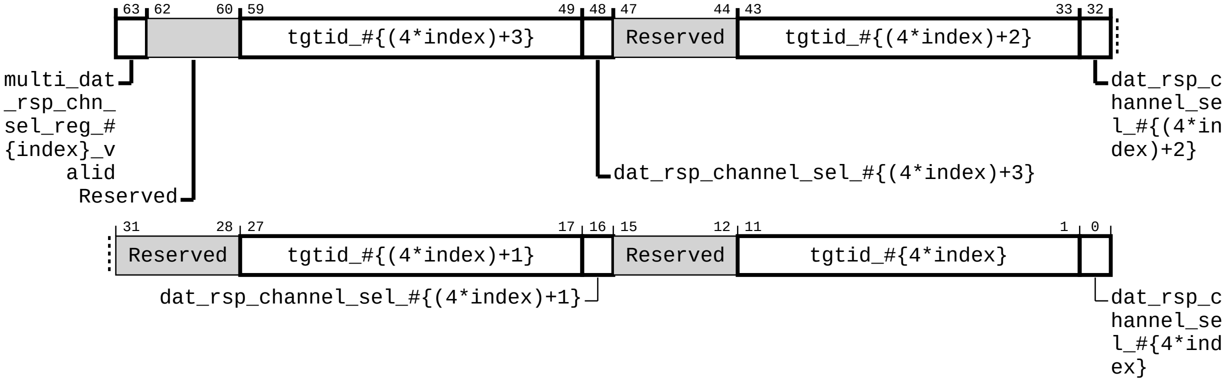

63
62
60
59
49
48
47
44
43
33
32
tgtid_#{(4*index)+3}
Reserved
tgtid_#{(4*index)+2}
multi_dat_rsp_chn_sel_reg_#{index}_valid
Reserved
dat_rsp_channel_sel_#{(4*index)+3}
dat_rsp_channel_sel_#{(4*index)+2}
31
28
27
17
16
15
12
11
1
0
Reserved
tgtid_#{(4*index)+1}
Reserved
tgtid_#{4*index}
dat_rsp_channel_sel_#{(4*index)+1}
dat_rsp_channel_sel_#{4*index}

Table 5-401: por_mxp_multi_dat_rsp_chn_sel_0-15 attributes

Bits	Name	Description	Type	Reset
[63]	multi_dat_rsp_chn_sel_reg_#{index}_valid	Indicates that dual DAT and RSP channel configured for the targets specified in this register.	RW	1'b0
[62:60]	Reserved	Reserved	RO	-
[59:49]	tgtid_#{(4*index)+3}	11-bit target ID associated with the corresponding dat_rsp_channel_sel field. This field is used in the Lookup Table (LUT) to determine which DAT/RSP channel the flit must be routed to for this target.	RW	11'b0
[48]	dat_rsp_channel_sel_#{(4*index)+3}	DAT/RSP channel select: 1: DAT/RSP channel 1 is selected 0: DAT/RSP channel 0 is selected	RW	1'b0
[47:44]	Reserved	Reserved	RO	-
[43:33]	tgtid_#{(4*index)+2}	11-bit target ID associated with the corresponding dat_rsp_channel_sel field. This field is used in the LUT to determine which DAT/RSP channel the flit must be routed to for this target.	RW	11'b0
[32]	dat_rsp_channel_sel_#{(4*index)+2}	DAT/RSP channel select: 1: DAT/RSP channel 1 is selected 0: DAT/RSP channel 0 is selected	RW	1'b0
[31:28]	Reserved	Reserved	RO	-
[27:17]	tgtid_#{(4*index)+1}	11-bit target ID associated with the corresponding dat_rsp_channel_sel field. This field is used in the LUT to determine which DAT/RSP channel the flit must be routed to for this target.	RW	11'b0
[16]	dat_rsp_channel_sel_#{(4*index)+1}	DAT/RSP channel select: 1: DAT/RSP channel 1 is selected 0: DAT/RSP channel 0 is selected	RW	1'b0
[15:12]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[11:1]	tgtid_#{4*index}	11-bit target ID associated with the corresponding dat_rsp_channel_sel field. This field is used in the LUT to determine which DAT/RSP channel the flit must be routed to for this target.	RW	11'b0
[0]	dat_rsp_channel_sel_#{4*index}	DAT/RSP channel select: 1: DAT/RSP channel 1 is selected 0: DAT/RSP channel 0 is selected	RW	1'b0

5.3.14.43 por_mxp_multi_dat_rsp_chn_ctrl

Functions as the control register when using the dual DAT/RSP feature per target for XP.

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_dat_rsp_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

Figure 5-386: por_mxp_multi_dat_rsp_chn_ctrl

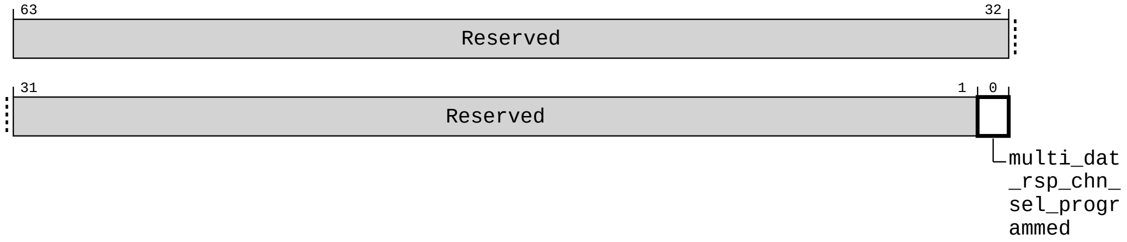


Table 5-402: por_mxp_multi_dat_rsp_chn_ctrl attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	multi_dat_rsp_chn_sel_programmed	Indicates that dual DAT and RSP channel configured for all the targets specified in the channel select registers.	RW	1'b0

5.3.14.44 por_mxp_xy_override_sel_0-7

There are 8 iterations of this register, parameterized by the index from 0 to 7. Functions as the source-target pair whose XY route path can be overridden per XP, when using the non-XY routing feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC90 + (8 × #[0, 1, ... 7])

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

Figure 5-387: por_mxp_xy_override_sel_0-7

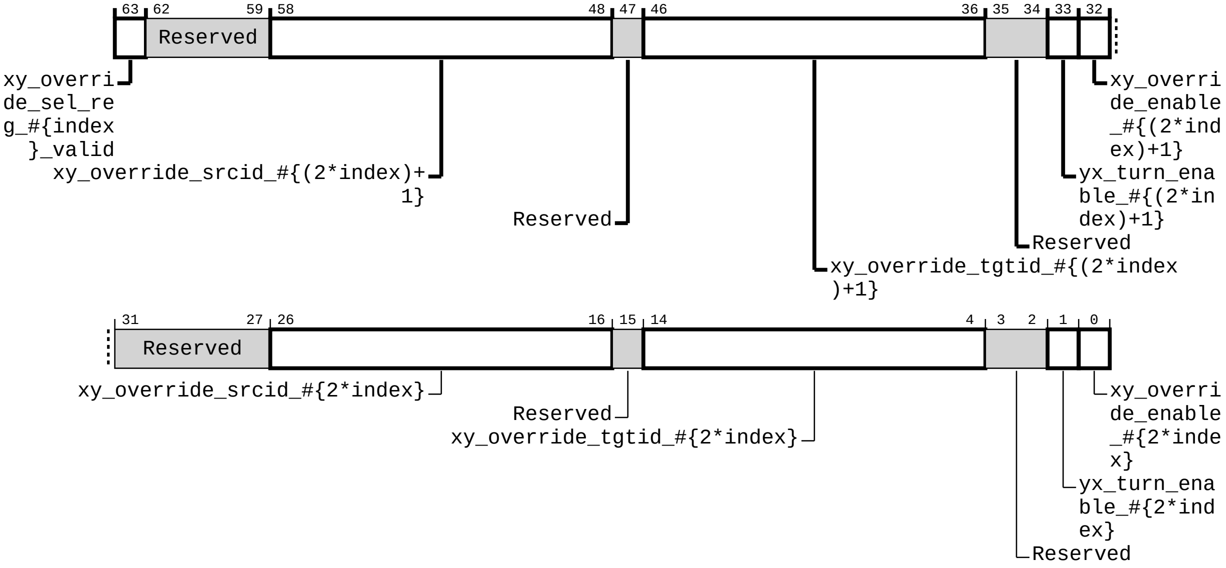

63
62
59
58
48
47
46
36
35
34
33
32
Reserved
xy_override_sel_reg_#{index}_valid
xy_override_srcid_#{(2*index)+1}
Reserved
xy_override_tgtid_#{(2*index)+1}
Reserved
yx_turn_enable_#{(2*index)+1}
xy_override_enable_#{(2*index)+1}
31
27
26
16
15
14
4
3
2
1
0
Reserved
xy_override_srcid_#{2*index}
Reserved
xy_override_tgtid_#{2*index}
Reserved
yx_turn_enable_#{2*index}
xy_override_enable_#{2*index}

Table 5-403: 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 XY route path can be overridden 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 Lookup Table (LUT) to determine if XY route for the associated source-target pair needs to be overridden.	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 LUT to determine if XY route for the associated source-target pair needs to be overridden.	RW	11'b0
[35:34]	Reserved	Reserved	RO	-
[33]	yx_turn_enable_#{(2*index)+1}	YX turn enable: 1: YX turn enabled for associated source-target pair 0: YX turn disabled	RW	1'b0
[32]	xy_override_enable_#{(2*index)+1}	XY route override enable: 1: XY route override enabled for associated source-target pair 0: XY route override disabled	RW	1'b0
[31:27]	Reserved	Reserved	RO	-
[26:16]	xy_override_srcid_#{2*index}	11-bit source ID associated with the XY override. This field is used in the LUT to determine if XY route for the associated source-target pair needs to be overridden.	RW	11'b0

Bits	Name	Description	Type	Reset
[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 LUT to determine if XY route for the associated source-target pair needs to be overridden.	RW	11'b0
[3:2]	Reserved	Reserved	RO	-
[1]	yx_turn_enable_#{2*index}	YX turn enable: 1 - YX turn enabled for associated source-target pair 0 - YX turn disabled	RW	1'b0
[0]	xy_override_enable_#{2*index}	XY route override enable: 1: XY route override enabled for the associated source-target pair 0: XY route override disabled	RW	1'b0

5.3.15 HN-F register descriptions

This section lists the HN-F registers.

5.3.15.1 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-388: por_hnf_node_info

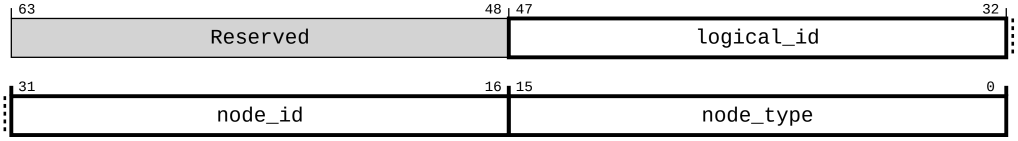


Table 5-404: por_hnf_node_info attributes

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-650 node type identifier	RO	16'h0005

5.3.15.2 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-389: por_hnf_child_info

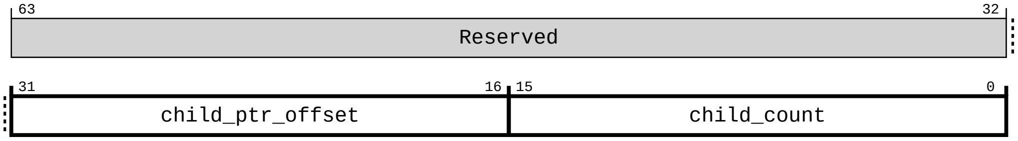


Table 5-405: por_hnf_child_info attributes

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. This value is used in the discovery process.	RO	16'b0

5.3.15.3 por_hnf_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

Figure 5-390: por_hnf_secure_register_groups_override

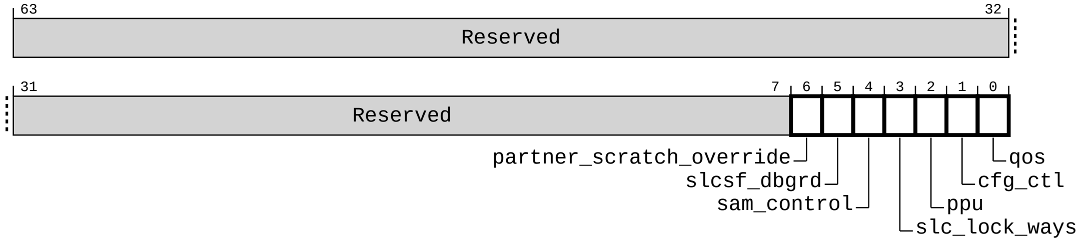


Table 5-406: por_hnf_secure_register_groups_override attributes

Bits	Name	Description	Type	Reset
[63:7]	Reserved	Reserved	RO	-
[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 and 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, por_hnf_cfg_ctl	RW	1'b0
[0]	qos	Allows Non-secure access to Secure QoS registers	RW	1'b0

5.3.15.4 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-391: por_hnf_unit_info

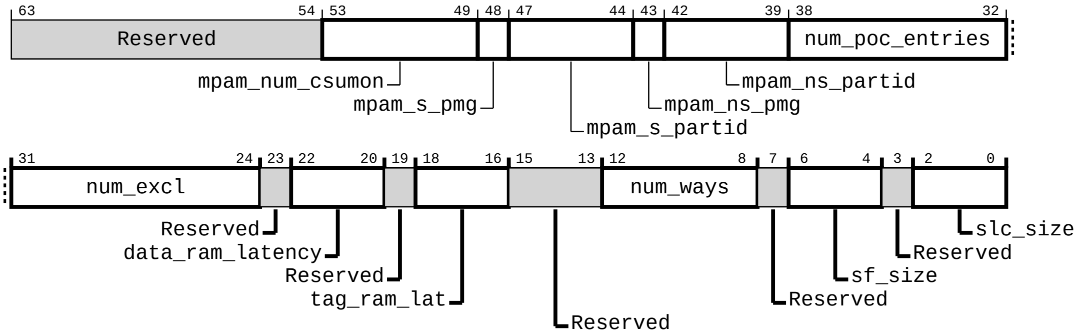


Table 5-407: por_hnf_unit_info attributes

Bits	Name	Description	Type	Reset
[63:54]	Reserved	Reserved	RO	-
[53:49]	mpam_num_csumon	Number of Cache Storage Usage (CSU) monitors for MPAM	RO	Configuration dependent
[48]	mpam_s_pmg	MPAM Secure supported PMGs: 1'b0: 1 S PMG 1'b1: 2 S PMG	RO	-
[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	-

Bits	Name	Description	Type	Reset
[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	-
[38:32]	num_poc_entries	Number of POCQ entries	RO	Configuration dependent
[31:24]	num_excl	Number of exclusive monitors	RO	-
[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]	num_ways	Number of cache ways in the SLC	RO	-
[7]	Reserved	Reserved	RO	-
[6:4]	sf_size	SF size: 3'b000: 512KB 3'b001: 1MB 3'b010: 2MB 3'b011: 4MB 3'b100: 8MB	RO	-
[3]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[2:0]	slc_size	SLC size: 3'b000: No SLC 3'b001: 128KB 3'b010: 256KB 3'b011: 512KB 3'b100: 1MB 3'b101: 2MB 3'b110: 3MB 3'b111: 4MB	RO	-

5.3.15.5 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-392: por_hnf_unit_info_1

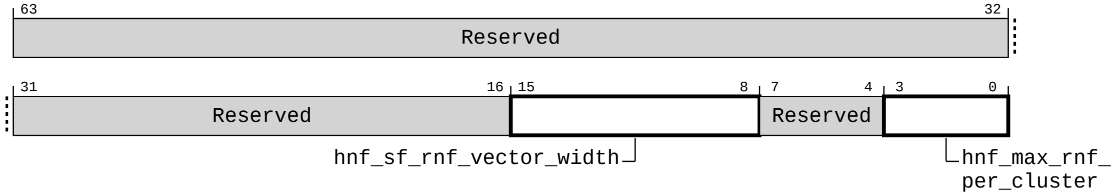


Table 5-408: por_hnf_unit_info_1 attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	hnf_sf_rnf_vector_width	Total number of bits in the RN-F tracking vector in the Snoop Filter (SF). Total SF_VEC_WIDTH = (TOTAL_RNF/HNF_MAX_CLUSTER_PARAM) + HNF_SF_ADD_VECTOR_WIDTH	RO	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3:0]	hnf_max_rnf_per_cluster	Indicates the maximum number of RN-Fs in a single cluster	RO	Configuration dependent

5.3.15.6 por_hnf_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

por_hnf_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

Figure 5-393: por_hnf_cfg_ctl

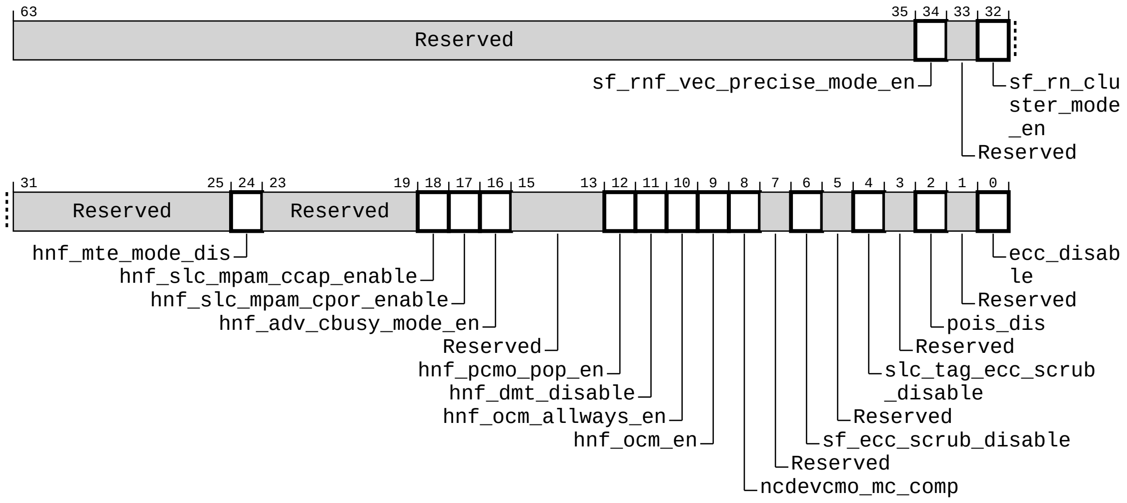


Table 5-409: por_hnf_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34]	sf_rnf_vec_precise_mode_en	Enables precise RN-F vector tracking in SF clustered mode	RW	1'b0
[33]	Reserved	Reserved	RO	-
[32]	sf_rn_cluster_mode_en	Enables the SF clustering of the RN-F IDs using programmable registers	RW	1'b1
[31:25]	Reserved	Reserved	RO	-
[24]	hnf_mte_mode_dis	If set to 1, disables the MTE features in HN-F	RW	1'b0
[23:19]	Reserved	Reserved	RO	-
[18]	hnf_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
[17]	hnf_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]	hnf_adv_cbusy_mode_en	Enables the advanced features of HN-F CBusy handling	RW	1'b0
[15:13]	Reserved	Reserved	RO	-
[12]	hnf_pcmo_pop_en	If this bit is set to 1, terminates PCMO in HN-F	RW	1'b0
[11]	hnf_dmt_disable	If set, disables DMT	RW	1'b0

Bits	Name	Description	Type	Reset
[10]	hnf_ocm_allways_en	Enables all SLC ways with OCM	RW	1'b0
[9]	hnf_ocm_en	Enables region locking with OCM support	RW	1'b0
[8]	ncdevcmo_mc_comp	If set, disables HN-F completion NOTE: If set, HN-F sends completion for the following transactions received after completion from SN: 1. Non-cacheable WriteNoSnp 2. Device WriteNoSnp 3. Cache Maintenance Operations (CMOs) CONSTRAINT: When this bit is set, the dis_ncwr_stream fields of the por_rni_cfg_ctl and por_rnd_cfg_ctl registers must also be set.	RW	1'b0
[7]	Reserved	Reserved	RO	-
[6]	sf_ecc_scrub_disable	If set, disables SF tag single-bit ECC error scrubbing	RW	1'b0
[5]	Reserved	Reserved	RO	-
[4]	slc_tag_ecc_scrub_disable	If set, disables SLC tag single-bit ECC error scrubbing	RW	1'b0
[3]	Reserved	Reserved	RO	-
[2]	pois_dis	If set, disables parity error data poison	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	ecc_disable	If set, disables SLC and SF ECC generation and detection	RW	1'b0

5.3.15.7 por_hnf_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

Figure 5-394: por_hnf_aux_ctl

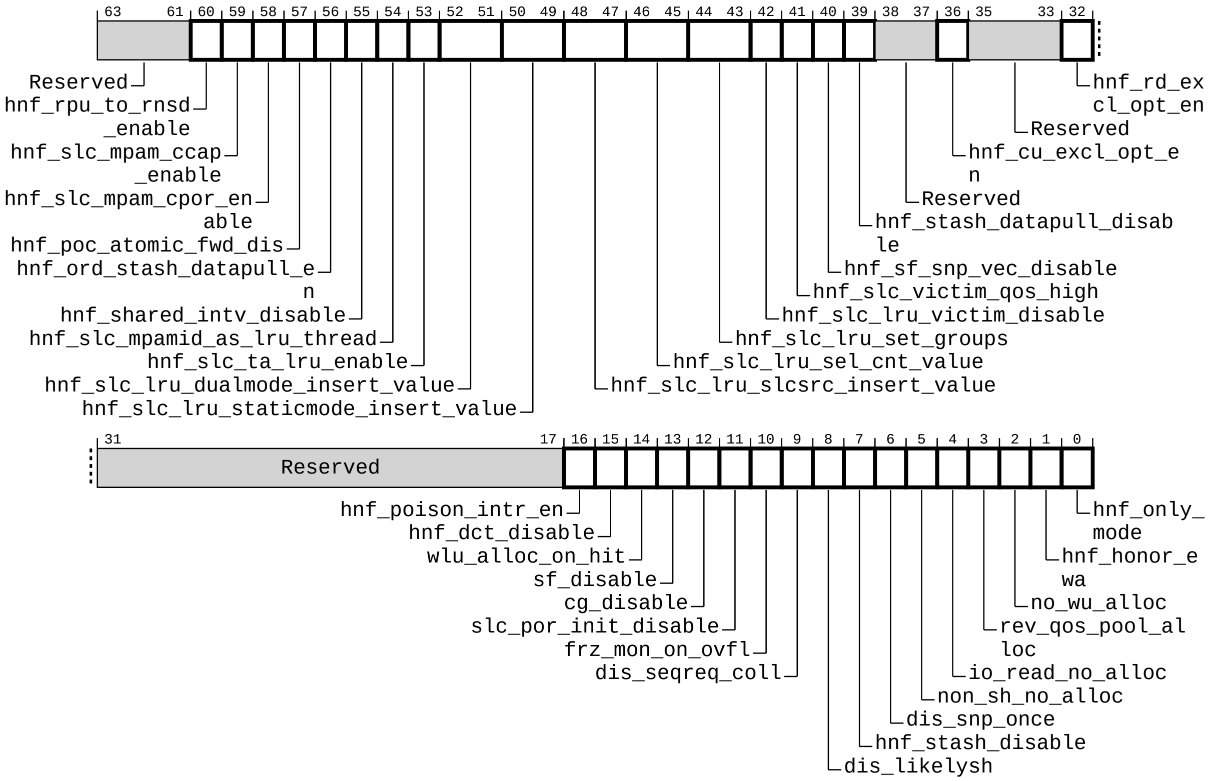


Table 5-410: por_hnf_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60]	hnf_rpu_to_rnsd_enable	Enables HN-F to treat ReadPrefUnique operations as ReadNotSharedDirty	RW	1'b0
[59]	hnf_slc_mpam_ccap_enable	Enable MPAM cache capcity partitioning for SLC: 1'b1: Cache capcity partitioning is enabled if supported in hardware. 1'b0: Cache capcity partitioning is disabled for SLC. Note: This bit moved to cfg_ctl and will be removed in a future version of ICN. Note: If MPAM is disabled at build time, this bit has no meaning.	RW	1'b0

Bits	Name	Description	Type	Reset
[58]	hnf_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 a future version of ICN. Note: If MPAM is disabled at build time, this bit has no meaning.	RW	1'b0
[57]	hnf_poc_atomic_fwd_dis	Disable atomic data forwarding in POCQ	RW	1'b0
[56]	hnf_ord_stash_datapull_en	Enables stash DataPull for ordered write stash requests	RW	Configuration dependent
[55]	hnf_shared_intv_disable	Disables snoop requests to CHI-B RN-F with shared copy	RW	Configuration dependent
[54]	hnf_slc_mpamid_as_lru_thread	Use MPAM PARTID as ThreadID for Thread Aware enhanced Least Recently Used (eLRU): 1'b0: ThreadID is based on LPID + LID for Thread Aware eLRU. 1'b1: ThreadID is based on MPAM PARTID + NS for Thread Aware eLRU. Note: MPAM PARTID is used only if MPAM is enabled.	RW	1'b0
[53]	hnf_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 ignored.	RW	1'b0
[52:51]	hnf_slc_lru_dualmode_insert_value	Insertion value for dual mode eLRU Note: Default is 2'b11.	RW	2'b11
[50:49]	hnf_slc_lru_staticmode_insert_value	Insertion value for static mode eLRU Note: Default is 2'b10.	RW	2'b10
[48:47]	hnf_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]	hnf_slc_lru_sel_cnt_value	Selection counter value for eLRU to determine which group policy is more effective: 2'b00: Selection counter is an 8-bit range. The upper limit is 255 and the middle point is 128. 2'b01: Selection counter is a 9-bit range. The upper limit is 511 and the middle point is 256. 2'b10: Selection counter is a 10-bit range. The upper limit is 1023 and the middle point is 512. 2'b11: Selection counter is an 11-bit range. The upper limit is 2047 and the middle point is 1024. Note: Default is 10-bit with counter reset to a value of 512.	RW	2'b10
[44:43]	hnf_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 is only one set per group.	RW	2'b01
[42]	hnf_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]	hnf_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]	hnf_sf_snp_vec_disable	If set, disables SF snoop vector	RW	1'b0
[39]	hnf_stash_datapull_disable	If set, disables HN-F stash data pull support	RW	1'b0
[38:37]	Reserved	Reserved	RO	-
[36]	hnf_cu_excl_opt_en	CleanUnique exclusive optimization enable	RW	Configuration dependent
[35:33]	Reserved	Reserved	RO	-
[32]	hnf_rd_excl_opt_en	ReadNotSharedDirty exclusive optimization enable	RW	1'b0
[31:17]	Reserved	Reserved	RO	-
[16]	hnf_poison_intr_en	Enables reporting an interrupt by HN-F when poison is detected at SLC	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[15]	hnf_dct_disable	If set, disables DCT	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]	hnf_stash_disable	Disables HN-F stash support	RW	Configuration dependent
[6]	dis_snp_once	If set, disables SnpOnce and converts to SnpShared	RW	Configuration dependent
[5]	non_sh_no_alloc	If set, disables SLC allocation for non-shareable cacheable transactions	RW	1'b0
[4]	io_read_no_alloc	If set, disables ReadOnce and ReadNoSnp allocation in SLC from RN-Is	RW	1'b0
[3]	rev_qos_pool_alloc	Reverses QoS pool allocation algorithm	RW	1'b0
[2]	no_wu_alloc	If set, disables WriteUnique and WriteLineUnique allocations in SLC	RW	1'b0
[1]	hnf_honor_ewa	If set, postpones completion for writes where EWA = 0 in the request until HN-F receives completion from memory controller or SBSX	RW	1'b1
[0]	hnf_only_mode	Enables HN-F only mode. If set, disables SLC and SF.	RW	1'b0

5.3.15.8 por_hnf_r2_aux_ctl

Functions as the auxiliary control register for HN-F for CMN-650 R2 features.

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

Figure 5-395: por_hnf_r2_aux_ctl

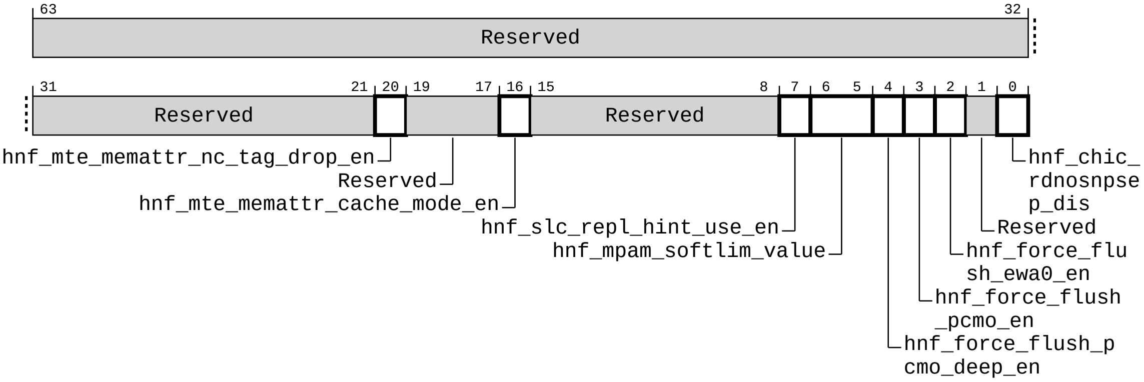


Table 5-411: por_hnf_r2_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:21]	Reserved	Reserved	RO	-
[20]	hnf_mte_memattr_nc_tag_drop_en	If set to 1, enables HN-F to drop any dirty tags for Non-Cacheable memory	RW	1'b0
[19:17]	Reserved	Reserved	RO	-
[16]	hnf_mte_memattr_cache_mode_en	If set to 1'b1, enables HN-F to convert non-cacheable requests to cacheable if MTE tags are required	RW	1'b1
[15:8]	Reserved	Reserved	RO	-
[7]	hnf_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'b0
[6:5]	hnf_mpam_softlim_value	Soft limit value for MPAM capacity partitioning: 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: The default value is 3.13% below hardlimit. If the CMAX value set is at or below 12.5%, the soft limit is ignored.	RW	2'b01
[4]	hnf_force_flush_pcmo_deep_en	Makes PCMO request for SLC and SF flush generate SN writes as Deep PCMO. CONSTRAINT: hnf_force_flush_pcmo_deep_en is valid only if hnf_force_flush_pcmo_en bit is set. CONSTRAINT: This bit can be set only if all SNs in the system support the deep attribute.	RW	1'b0
[3]	hnf_force_flush_pcmo_en	Generate PCMO request for SLC and SF flush generated SN writes	RW	1'b0
[2]	hnf_force_flush_ewa0_en	Force SLC and SF flush to use EWA 0 for SN writes	RW	1'b0
[1]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[0]	hnf_chic_rdnosnpsep_dis	Disables separation of Data and Comp in CHI-C mode	RW	1'b0

5.3.15.9 por_hnf_cbusy_limit_ctl

Completer Busy (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

Figure 5-396: por_hnf_cbusy_limit_ctl

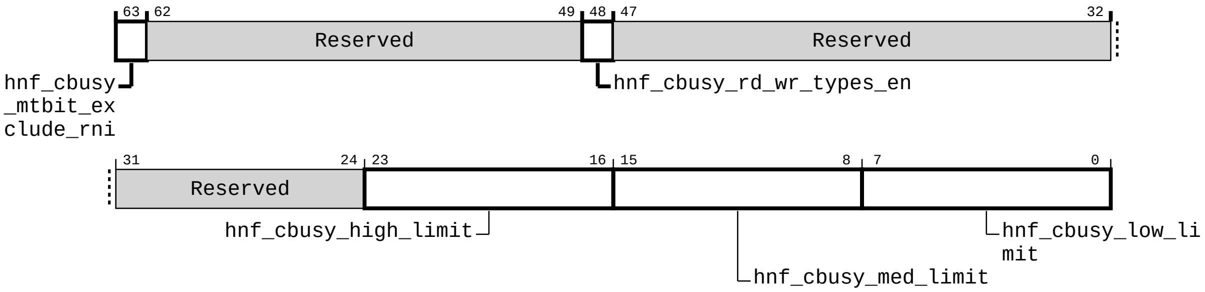


Table 5-412: por_hnf_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63]	hnf_cbusy_mtbit_exclude_rni	Exclude RN-I sources in multi-source mode	RW	1'b0
[62:49]	Reserved	Reserved	RO	-
[48]	hnf_cbusy_rd_wr_types_en	If set, CBusy for reads and writes are handled independently. The thresholds that are specified in this register are used for read request types in POCQ.	RW	1'b0

Bits	Name	Description	Type	Reset
[47:24]	Reserved	Reserved	RO	-
[23:16]	hnf_cbusy_high_limit	POCQ limit for CBusy high	RW	Configuration dependent
[15:8]	hnf_cbusy_med_limit	POCQ limit for CBusy medium	RW	Configuration dependent
[7:0]	hnf_cbusy_low_limit	POCQ limit for CBusy low	RW	Configuration dependent

5.3.15.10 por_hnf_ppu_pwpr

Functions as the power policy register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C00

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-397: por_hnf_ppu_pwpr

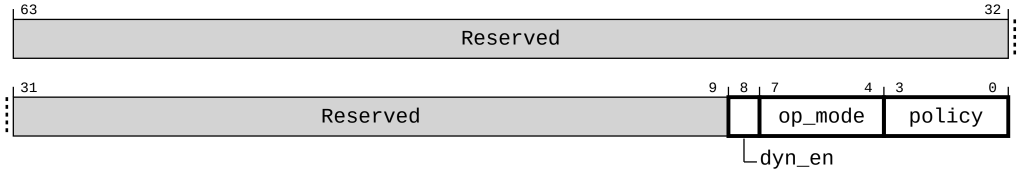


Table 5-413: por_hnf_ppu_pwpr attributes

Bits	Name	Description	Type	Reset
[63:9]	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

5.3.15.11 por_hnf_ppu_pwsr

Provides power status information for HN-F.

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

Figure 5-398: por_hnf_ppu_pwsr

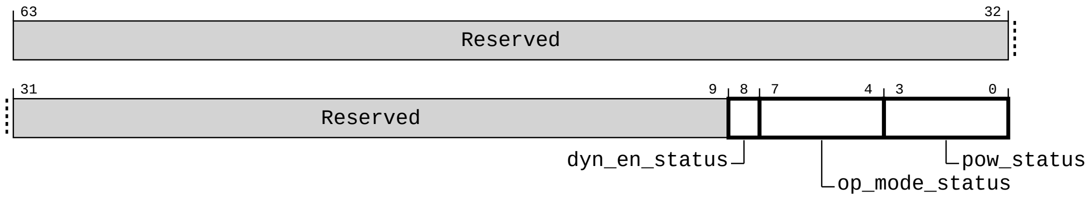


Table 5-414: por_hnf_ppu_pwsr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	dyn_en_status	Dynamic transition status	RO	1'b0
[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

5.3.15.12 por_hnf_ppu_misr

Functions as the power miscellaneous input current status register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1C14

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-399: por_hnf_ppu_misr

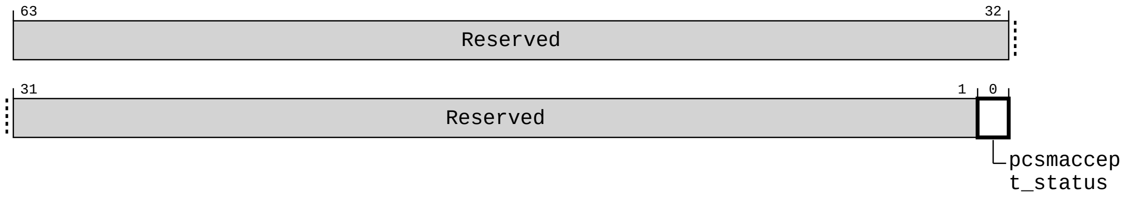


Table 5-415: por_hnf_ppu_misr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	pcsmaccept_status	HN-F RAM PCSMACCEPT status	RO	1'b0

5.3.15.13 por_hnf_ppu_idr0

Provides identification information for the HN-F PPU.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2BB0

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-400: por_hnf_ppu_idr0

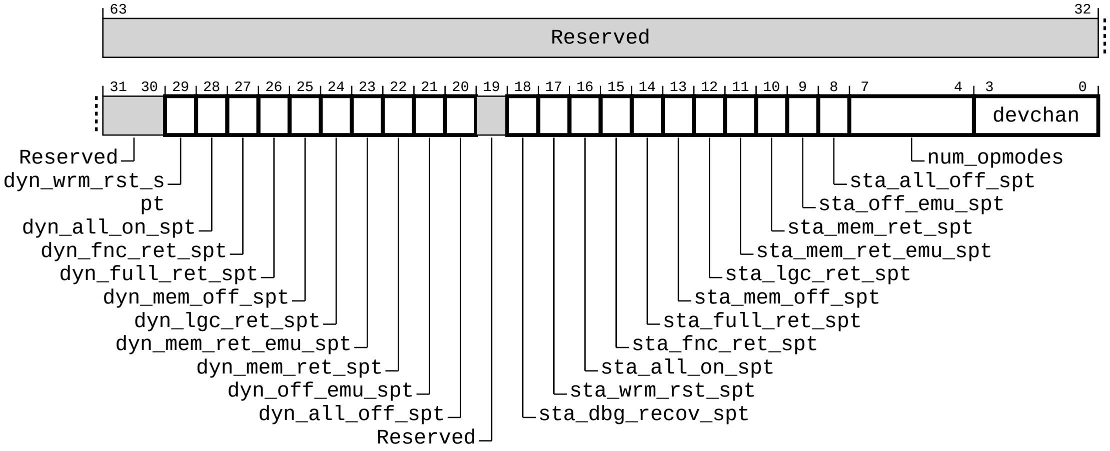


Table 5-416: por_hnf_ppu_idr0 attributes

Bits	Name	Description	Type	Reset
[63:30]	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

Bits	Name	Description	Type	Reset
[7:4]	num_opmodes	Number of operational modes	RO	4'b0100
[3:0]	devchan	Number of device interface channels	RO	1'b0

5.3.15.14 por_hnf_ppu_idr1

Provides identification information for the HN-F PPU.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2BB4

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-401: por_hnf_ppu_idr1

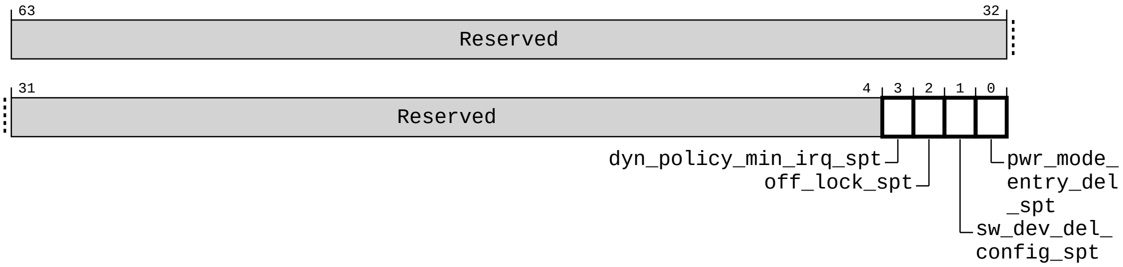


Table 5-417: por_hnf_ppu_idr1 attributes

Bits	Name	Description	Type	Reset
[63:4]	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

Bits	Name	Description	Type	Reset
[0]	pwr_mode_entry_del_spt	Power mode entry delay support	RO	1'b0

5.3.15.15 por_hnf_ppu_iidr

Functions as the power implementation identification register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2BC8

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-402: por_hnf_ppu_iidr

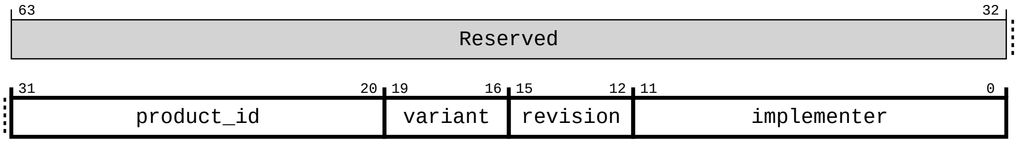


Table 5-418: por_hnf_ppu_iidr attributes

Bits	Name	Description	Type	Reset
[63:32]	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

5.3.15.16 por_hnf_ppu_aidr

Functions as the power architecture identification register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2BCC

Type

RO

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-403: por_hnf_ppu_aidr

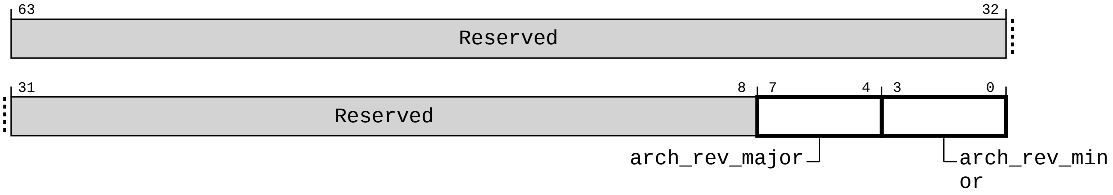


Table 5-419: por_hnf_ppu_aidr attributes

Bits	Name	Description	Type	Reset
[63:8]	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

5.3.15.17 por_hnf_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

por_hnf_secure_register_groups_override.ppu

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-404: por_hnf_ppu_dyn_ret_threshold

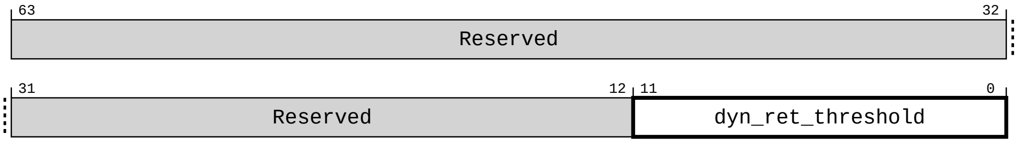


Table 5-420: por_hnf_ppu_dyn_ret_threshold attributes

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

5.3.15.18 por_hnf_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

por_hnf_secure_register_groups_override.qos

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-405: por_hnf_qos_band

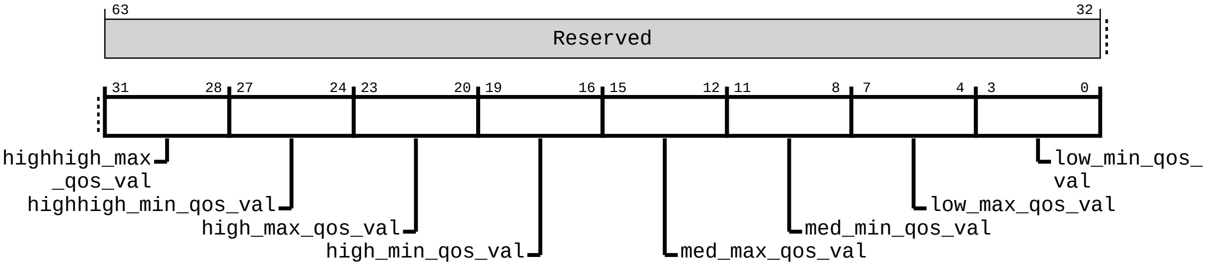


Table 5-421: por_hnf_qos_band attributes

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
[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

5.3.15.19 por_hnf_qos_reservation

Controls POCQ maximum occupancy counts for each QoS class, HighHigh, High, Medium, and Low.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA88

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.qos

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

Figure 5-406: por_hnf_qos_reservation

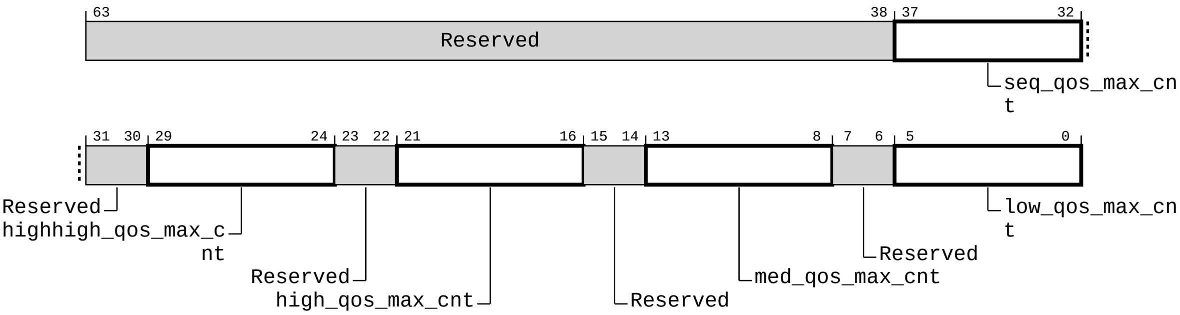


Table 5-422: por_hnf_qos_reservation attributes

Bits	Name	Description	Type	Reset
[63:38]	Reserved	Reserved	RO	-
[37:32]	seq_qos_max_cnt	Number of entries reserved for SF evictions in POCQ. CONSTRAINT: Maximum number is two entries.	RW	6'h1
[31:30]	Reserved	Reserved	RO	-
[29:24]	highhigh_qos_max_cnt	Maximum number of HighHigh QoS class occupancy. CONSTRAINT: Minimum is two entries.	RW	Configuration dependent
[23:22]	Reserved	Reserved	RO	-
[21:16]	high_qos_max_cnt	Maximum number of High QoS class occupancy. CONSTRAINT: Minimum is two entries.	RW	Configuration dependent
[15:14]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[13:8]	med_qos_max_cnt	Maximum number of Medium QoS class occupancy. CONSTRAINT: Minimum is two entries.	RW	Configuration dependent
[7:6]	Reserved	Reserved	RO	-
[5:0]	low_qos_max_cnt	Maximum number of Low QoS class occupancy. CONSTRAINT: Minimum is two entries.	RW	Configuration dependent

5.3.15.20 por_hnf_rn_starvation

Controls starvation counts for each QoS class. Determines static credit grantee selection.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA90

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.qos

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

Figure 5-407: por_hnf_rn_starvation

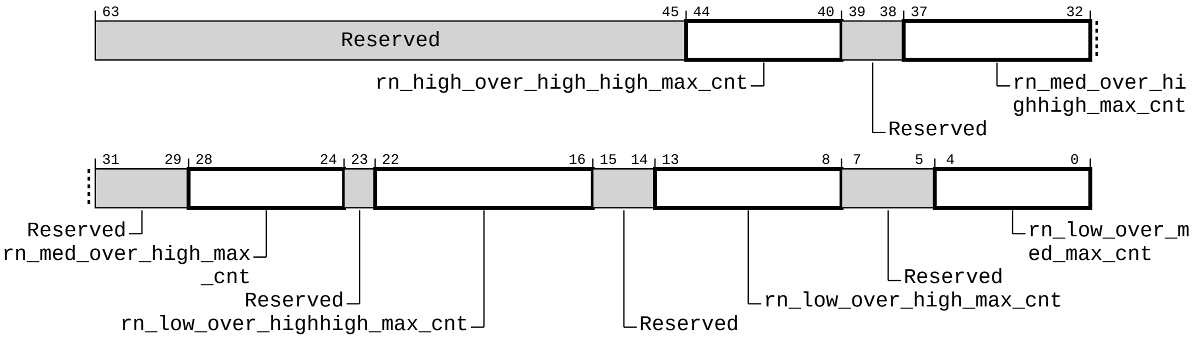


Table 5-423: por_hnf_rn_starvation attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:40]	rn_high_over_high_high_max_cnt	Maximum number of consecutive instances where HighHigh QoS class wins priority over High QoS class	RW	5'h1F
[39:38]	Reserved	Reserved	RO	-
[37:32]	rn_med_over_highhigh_max_cnt	Maximum number of consecutive instances where HighHigh QoS class wins priority over Medium QoS class	RW	6'h3F
[31:29]	Reserved	Reserved	RO	-
[28:24]	rn_med_over_high_max_cnt	Maximum number of consecutive instances where High QoS class wins priority over Medium QoS class	RW	5'h1F
[23]	Reserved	Reserved	RO	-
[22:16]	rn_low_over_highhigh_max_cnt	Maximum number of consecutive instances where HighHigh QoS class wins priority over Low QoS class	RW	7'h3F
[15:14]	Reserved	Reserved	RO	-
[13:8]	rn_low_over_high_max_cnt	Maximum number of consecutive instances where High QoS class wins priority over Low QoS class	RW	6'h3F
[7:5]	Reserved	Reserved	RO	-
[4:0]	rn_low_over_med_max_cnt	Maximum number of consecutive instances where Medium QoS class wins priority over Low QoS class	RW	5'h1F

5.3.15.21 por_hnf_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-408: por_hnf_errfr

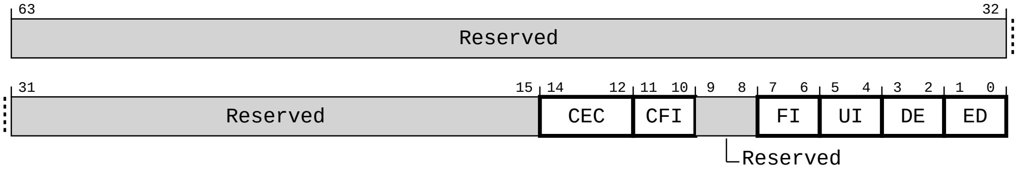


Table 5-424: por_hnf_errfr attributes

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_hnf_errmisc[39:32] 3'b100: Implements 16-bit error counter in por_hnf_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

5.3.15.22 por_hnf_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-409: por_hnf_errctlr

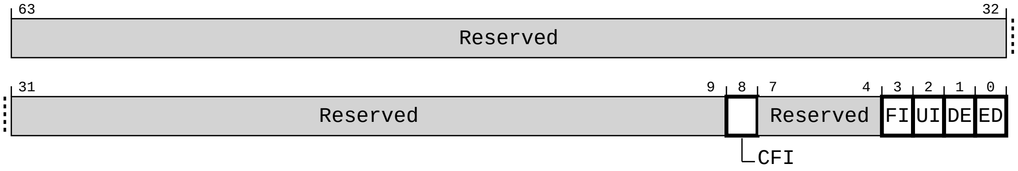


Table 5-425: por_hnf_errctlr attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_hnf_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_hnf_errfr.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_hnf_errfr.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_hnf_errfr.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_hnf_errfr.ED	RW	1'b0

5.3.15.23 por_hnf_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-410: por_hnf_errstatus

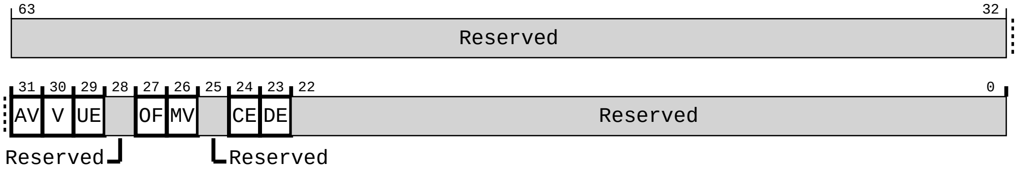


Table 5-426: por_hnf_errstatus attributes

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_hnf_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

Bits	Name	Description	Type	Reset
[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 slave 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, and CE fields	W1C	1'b0
[26]	MV	por_hnf_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	-

5.3.15.24 por_hnf_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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-411: por_hnf_erraddr

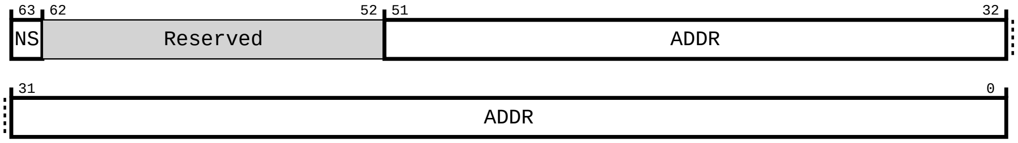


Table 5-427: por_hnf_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction: 1'b1: Non-secure transaction 1'b0: Secure transaction CONSTRAINT: por_hnf_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

5.3.15.25 por_hnf_errmisc

Functions as the miscellaneous error register. Contains miscellaneous information about deferred and 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

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-412: por_hnf_errmisc

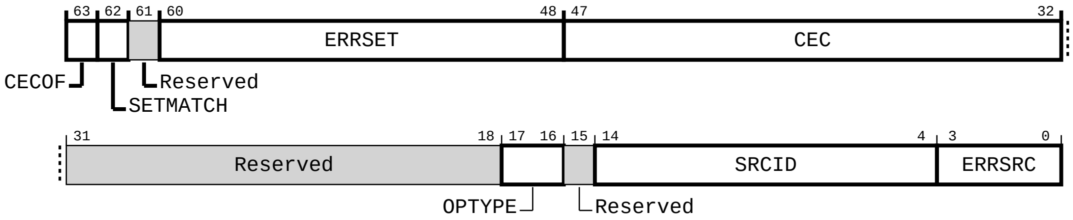


Table 5-428: por_hnf_errmisc attributes

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 or SF set address for ECC error	RW	13'b0
[47:32]	CEC	Corrected ECC error count	RW	16'b0
[31:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error operation 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

5.3.15.26 por_hnf_err_inj

Enables error injection and setup.

When enabled for a given source ID and logic processor ID, HN-F returns a slave 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 slave error is reported for cacheable read access where an SLC hit is the data source. No slave error or error interrupt is reported for cacheable read accesses where an 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

Figure 5-413: por_hnf_err_inj

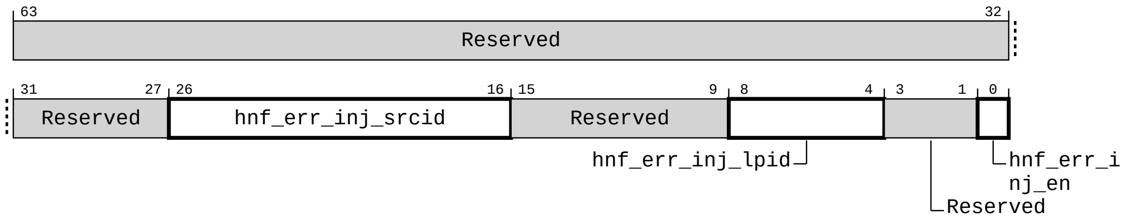


Table 5-429: por_hnf_err_inj attributes

Bits	Name	Description	Type	Reset
[63:27]	Reserved	Reserved	RO	-
[26:16]	hnf_err_inj_srcid	RN source ID for read access which results in a SLC miss. Does not report slave error or error to match error injection.	RW	11'h0
[15:9]	Reserved	Reserved	RO	-
[8:4]	hnf_err_inj_lpid	LPID used to match for error injection	RW	5'h0
[3:1]	Reserved	Reserved	RO	-
[0]	hnf_err_inj_en	Enables error injection and report	RW	1'b0

5.3.15.27 por_hnf_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

Figure 5-414: por_hnf_byte_par_err_inj

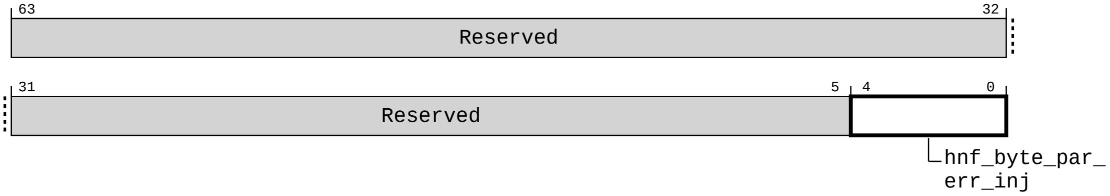


Table 5-430: por_hnf_byte_par_err_inj attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:0]	hnf_byte_par_err_inj	Specifies a byte lane. After this register is written, a byte parity error is injected in the specified byte lane on the next SLC hit. The error is injected in all data flits on specified byte (0-31)	WO	5'h0

5.3.15.28 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-415: por_hnf_errfr_NS

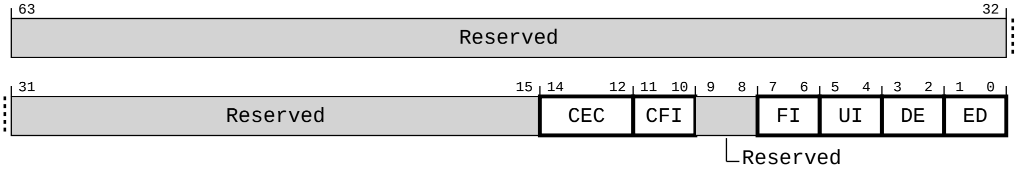


Table 5-431: por_hnf_errfr_NS attributes

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_hnf_errmisc_NS[39:32] 3'b100: Implements 16-bit error counter in por_hnf_errmisc_NS[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

5.3.15.29 por_hnf_errctlr_NS

Functions as the Non-secure error control register. Controls whether specific error-handling interrupts and error detection and 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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-416: por_hnf_errctlr_NS

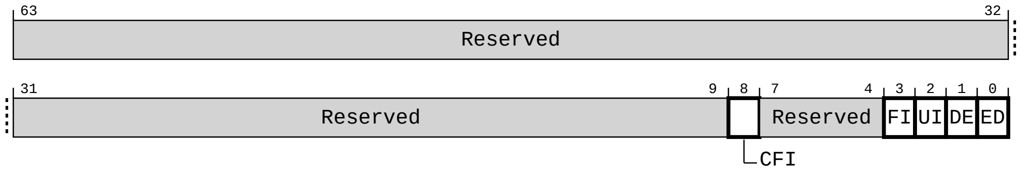


Table 5-432: por_hnf_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:9]	Reserved	Reserved	RO	-
[8]	CFI	Enables corrected error interrupt as specified in por_hnf_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_hnf_errfr_NS.FI	RW	1'b0
[2]	UI	Enables uncorrected error interrupt as specified in por_hnf_errfr_NS.UI	RW	1'b0
[1]	DE	Enables error deferment as specified in por_hnf_errfr_NS.DE	RW	1'b0
[0]	ED	Enables error detection as specified in por_hnf_errfr_NS.ED	RW	1'b0

5.3.15.30 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-417: por_hnf_errstatus_NS

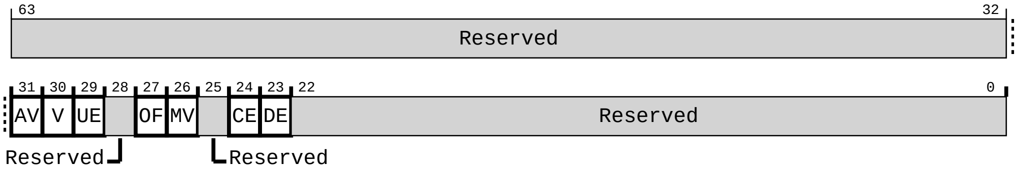


Table 5-433: por_hnf_errstatus_NS attributes

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_hnf_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 slave 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, and CE fields	W1C	1'b0
[26]	MV	por_hnf_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	-

5.3.15.31 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-418: por_hnf_erraddr_NS

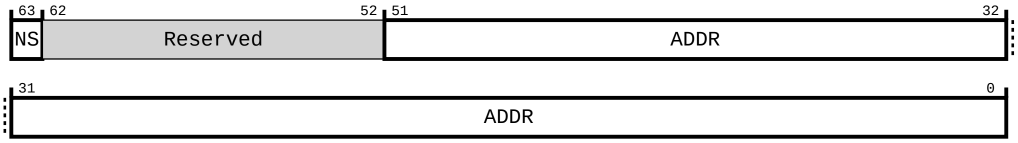


Table 5-434: por_hnf_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction: 1'b1: Non-secure transaction 1'b0: Secure transaction CONSTRAINT: por_hnf_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

5.3.15.32 por_hnf_errmisc_NS

Functions as the Non-secure miscellaneous error register. Contains miscellaneous information about deferred and 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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-419: por_hnf_errmisc_NS

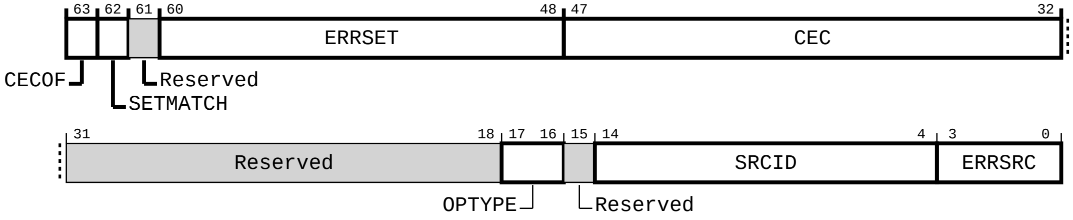


Table 5-435: por_hnf_errmisc_NS attributes

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 or SF set address for ECC error	RW	13'b0
[47:32]	CEC	Corrected ECC error count	RW	16'b0
[31:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error operation 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 operation 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

5.3.15.33 por_hnf_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

por_hnf_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

Figure 5-420: por_hnf_slc_lock_ways

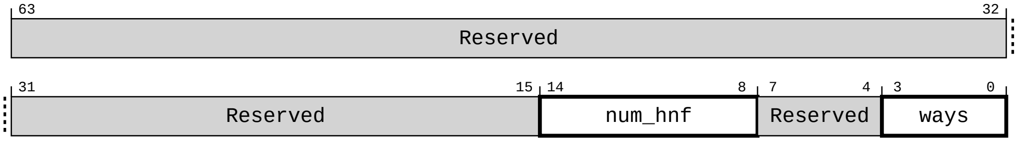


Table 5-436: por_hnf_slc_lock_ways attributes

Bits	Name	Description	Type	Reset
[63:15]	Reserved	Reserved	RO	-
[14:8]	num_hnf	Number of HN-Fs in Non-Uniform Memory Access (NUMA) region	RW	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3:0]	ways	Number of SLC ways locked: 1, 2, 4, 8, or 12	RW	4'b0

5.3.15.34 por_hnf_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

por_hnf_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

Figure 5-421: por_hnf_slc_lock_base0

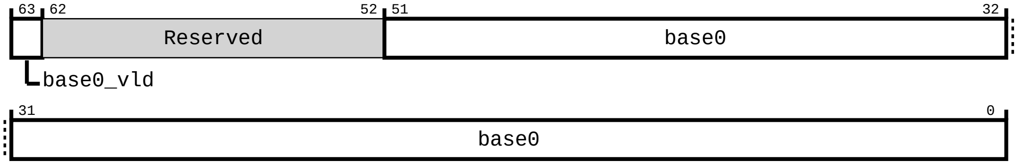


Table 5-437: por_hnf_slc_lock_base0 attributes

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

5.3.15.35 por_hnf_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

por_hnf_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

Figure 5-422: por_hnf_slc_lock_base1

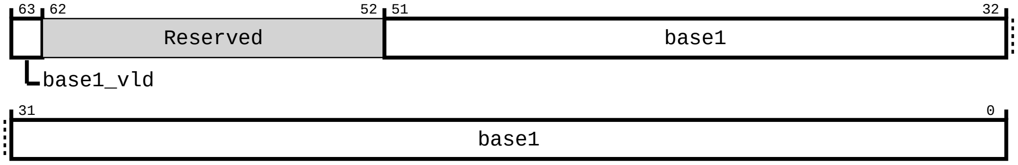


Table 5-438: por_hnf_slc_lock_base1 attributes

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

5.3.15.36 por_hnf_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

por_hnf_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

Figure 5-423: por_hnf_slc_lock_base2

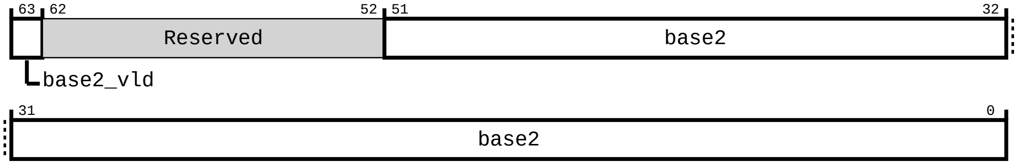


Table 5-439: por_hnf_slc_lock_base2 attributes

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

5.3.15.37 por_hnf_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

por_hnf_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

Figure 5-424: por_hnf_slc_lock_base3

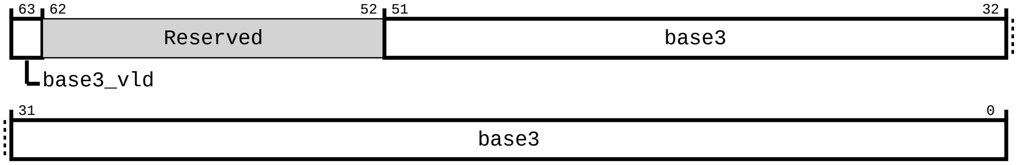


Table 5-440: por_hnf_slc_lock_base3 attributes

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

5.3.15.38 por_hnf_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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-425: por_hnf_rni_region_vec

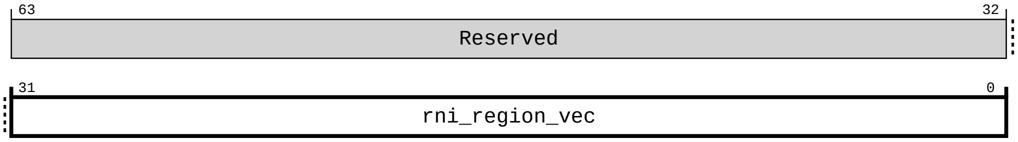


Table 5-441: por_hnf_rni_region_vec attributes

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

5.3.15.39 por_hnf_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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-426: por_hnf_rnd_region_vec

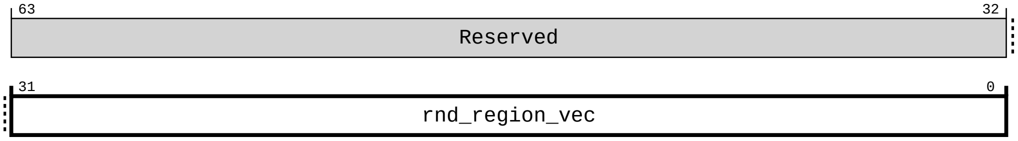


Table 5-442: por_hnf_rnd_region_vec attributes

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

5.3.15.40 por_hnf_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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-427: por_hnf_rnf_region_vec

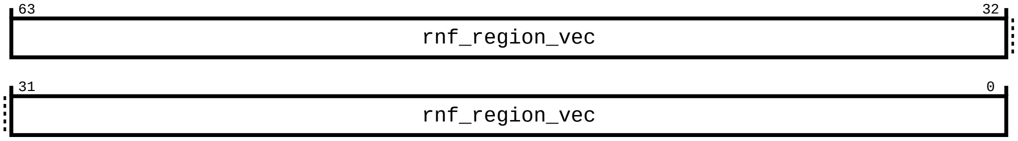


Table 5-443: por_hnf_rnf_region_vec attributes

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

5.3.15.41 por_hnf_rnf_region_vec1

Functions as the control register for RN-F source SLC way allocation for logical IDs 64-127.

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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-428: por_hnf_rnf_region_vec1

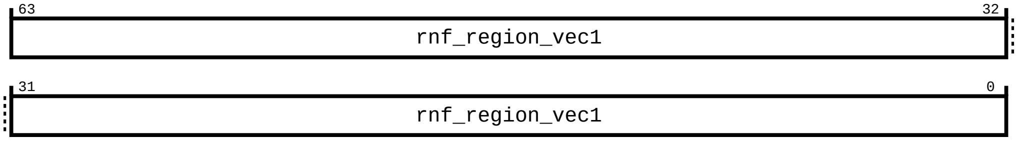


Table 5-444: por_hnf_rnf_region_vec1 attributes

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

5.3.15.42 por_hnf_slcway_partition0_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 0, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-429: por_hnf_slcway_partition0_rnf_vec

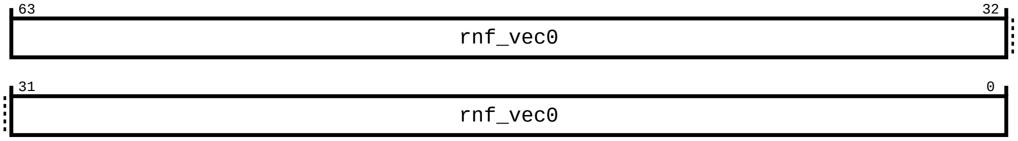


Table 5-445: por_hnf_slcway_partition0_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec0	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.43 por_hnf_slcway_partition1_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 1, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-430: por_hnf_slcway_partition1_rnf_vec

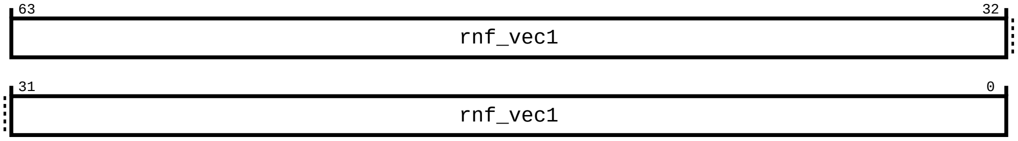


Table 5-446: por_hnf_slcway_partition1_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec1	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.44 por_hnf_slcway_partition2_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 2, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-431: por_hnf_slcway_partition2_rnf_vec

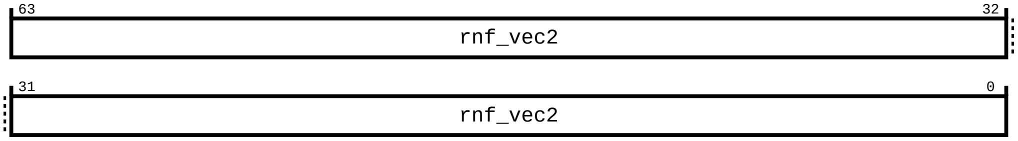


Table 5-447: por_hnf_slcway_partition2_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec2	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.45 por_hnf_slcway_partition3_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 3, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-432: por_hnf_slcway_partition3_rnf_vec

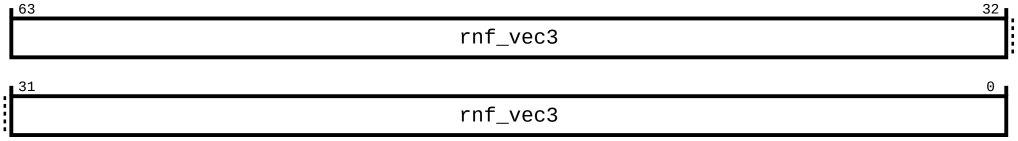


Table 5-448: por_hnf_slcway_partition3_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec3	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.46 por_hnf_slcway_partition0_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 0, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-433: por_hnf_slcway_partition0_rnf_vec1

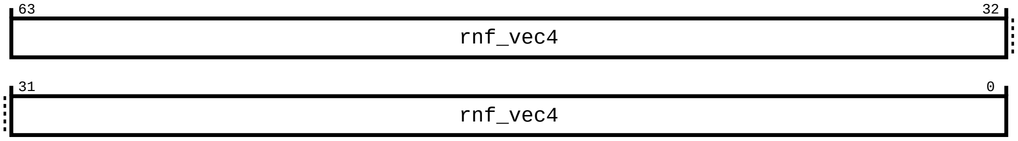


Table 5-449: por_hnf_slcway_partition0_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec4	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.47 por_hnf_slcway_partition1_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 1, in other words ways 4, 5, 6, and 7, for logical RN-F IDs 64-127.

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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-434: por_hnf_slcway_partition1_rnf_vec1

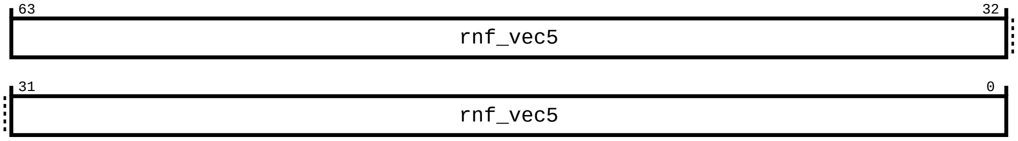


Table 5-450: por_hnf_slcway_partition1_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec5	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.48 por_hnf_slcway_partition2_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 2, in other words ways 8, 9, 10, and 11, for logical RN-F IDs 64-127.

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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-435: por_hnf_slcway_partition2_rnf_vec1

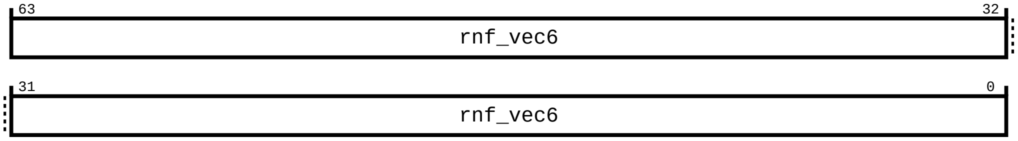


Table 5-451: por_hnf_slcway_partition2_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec6	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.49 por_hnf_slcway_partition3_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 3, in other words ways 12, 13, 14, and 15, for logical RN-F IDs 64-127.

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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-436: por_hnf_slcway_partition3_rnf_vec1

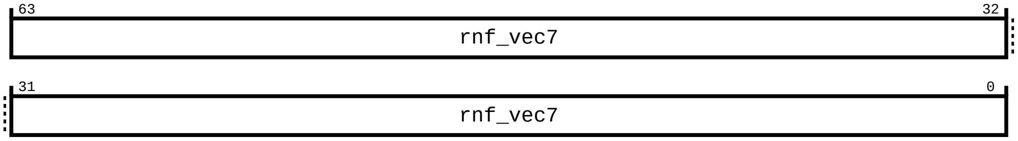


Table 5-452: por_hnf_slcway_partition3_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec7	Bit vector mask. Identifies which RN-F logical IDs can allocate.	RW	64'hFFFFFFFFFFFFFFFF

5.3.15.50 por_hnf_slcway_partition0_rni_vec

Functions as the control register for RN-Is that can allocate to partition 0, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-437: por_hnf_slcway_partition0_rni_vec

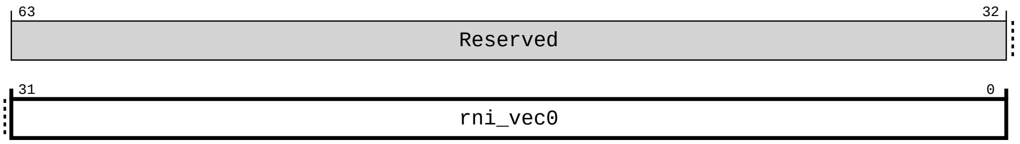


Table 5-453: por_hnf_slcway_partition0_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec0	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.51 por_hnf_slcway_partition1_rni_vec

Functions as the control register for RN-Is that can allocate to partition 1, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-438: por_hnf_slcway_partition1_rni_vec

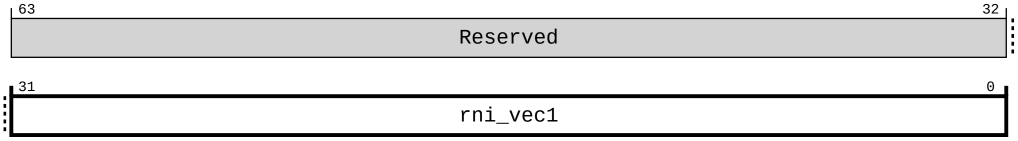


Table 5-454: por_hnf_slcway_partition1_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec1	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.52 por_hnf_slcway_partition2_rni_vec

Functions as the control register for RN-Is that can allocate to partition 2, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-439: por_hnf_slcway_partition2_rni_vec

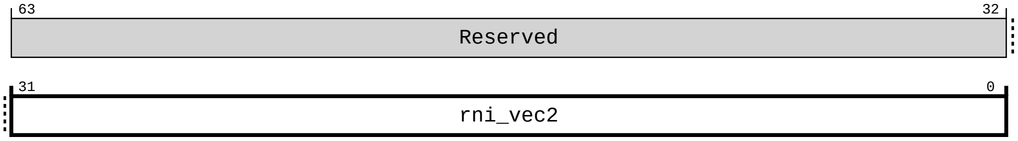


Table 5-455: por_hnf_slcway_partition2_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec2	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.53 por_hnf_slcway_partition3_rni_vec

Functions as the control register for RN-Is that can allocate to partition 3, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-440: por_hnf_slcway_partition3_rni_vec

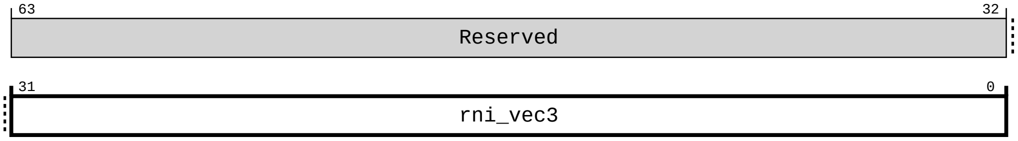


Table 5-456: por_hnf_slcway_partition3_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec3	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.54 por_hnf_slcway_partition0_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 0, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-441: por_hnf_slcway_partition0_rnd_vec

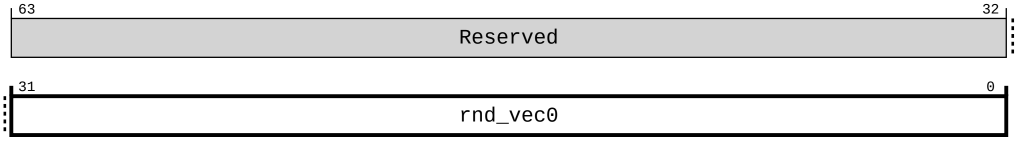


Table 5-457: por_hnf_slcway_partition0_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec0	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.55 por_hnf_slcway_partition1_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 1, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-442: por_hnf_slcway_partition1_rnd_vec

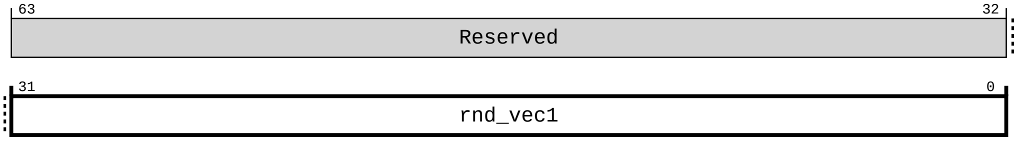


Table 5-458: por_hnf_slcway_partition1_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec1	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.56 por_hnf_slcway_partition2_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 2, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-443: por_hnf_slcway_partition2_rnd_vec

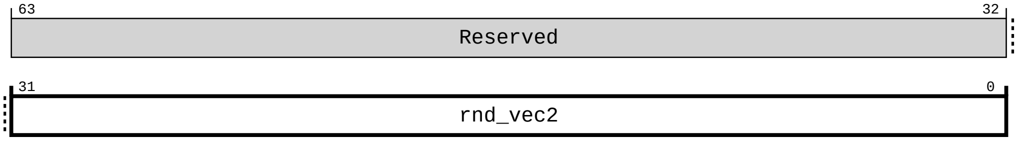


Table 5-459: por_hnf_slcway_partition2_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec2	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.57 por_hnf_slcway_partition3_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 3, in other words 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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-444: por_hnf_slcway_partition3_rnd_vec

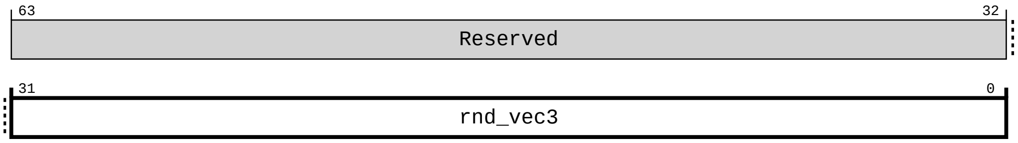


Table 5-460: por_hnf_slcway_partition3_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec3	Bit vector mask. Identifies which RN-I and RN-D logical IDs can allocate.	RW	32'hFFFFFFFF

5.3.15.58 por_hnf_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

por_hnf_secure_register_groups_override.slc_lock_ways

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-445: por_hnf_rn_region_lock

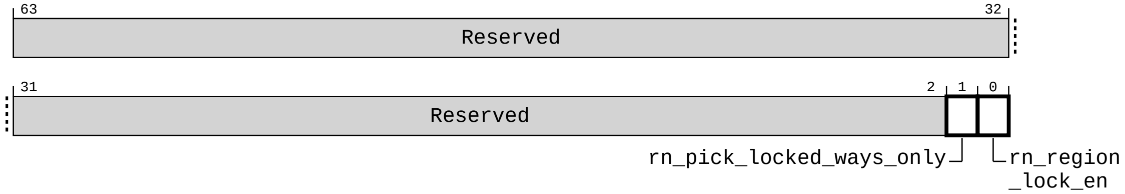


Table 5-461: por_hnf_rn_region_lock attributes

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 chooses all ways including locked 1'b1: Programmed RN only allocates 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

5.3.15.59 por_hnf_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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-446: por_hnf_sf_cxg_blocked_ways

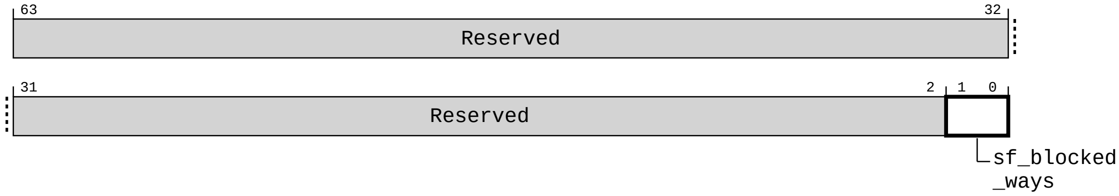


Table 5-462: por_hnf_sf_cxg_blocked_ways attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	sf_blocked_ways	Number of SF ways blocked for remote chips to use in CML mode (0, 4, 8, or 12): 2'b00: No ways are blocked. Local or remote RN-Fs can use all 16 SF ways. 2'b01: Lower four ways are blocked from remote RN-Fs. Only local RN-Fs can use ways 3-0. Local and remote RN-Fs can use ways 15-4. 2'b10: Lower eight ways are blocked from remote RN-Fs. Only local RN-Fs can use ways 7-0. Local and remote RN-Fs can use ways 15-8. 2'b11: Lower 12 ways are blocked from remote RN-Fs. Only local RN-Fs can use ways 11-0. Local and remote RN-Fs can use ways 15-12.	RW	2'b00

5.3.15.60 por_hnf_cxg_ha_metadata_exclusion_list

Functions as the control register to identify CXHAs which do 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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-447: por_hnf_cxg_ha_metadata_exclusion_list

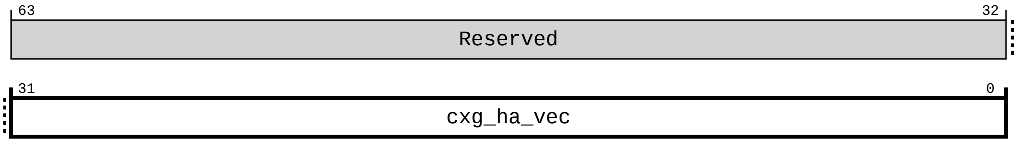


Table 5-463: por_hnf_cxg_ha_metadata_exclusion_list attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	cxg_ha_vec	Bit vector mask. Identifies which CXHA logical IDs do not support metadata.	RW	32'h00000000

5.3.15.61 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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-448: hn_sam_hash_addr_mask_reg

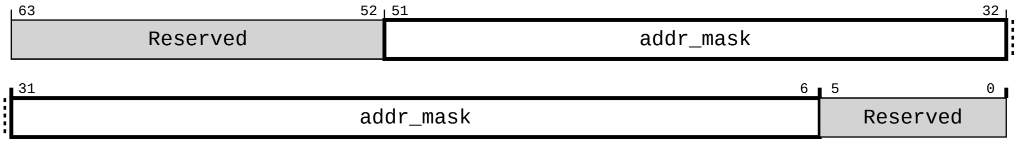


Table 5-464: hn_sam_hash_addr_mask_reg attributes

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	-

5.3.15.62 hn_sam_region_cmp_addr_mask_reg

Configures the address mask that is applied before memory region comparison.

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. This register can be modified only with prior written permission from Arm.

Bit descriptions

Figure 5-449: hn_sam_region_cmp_addr_mask_reg

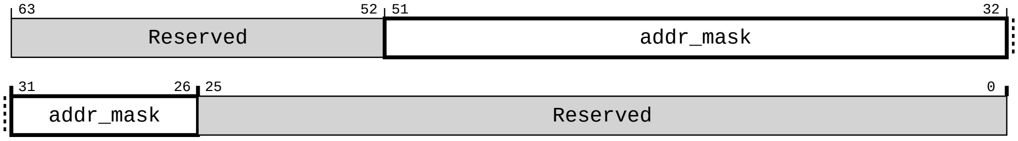


Table 5-465: hn_sam_region_cmp_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:26]	addr_mask	Address mask applied before memory region comparison	RW	26'h3FFFFFF
[25:0]	Reserved	Reserved	RO	-

5.3.15.63 por_hnf_sam_control

Configures HN-F SAM. All top_address_bit fields must be between bits [47:28] of the address. top_address_bit2 > top_address_bit1 > top_address_bit0. You must configure this register to match the 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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-450: por_hnf_sam_control

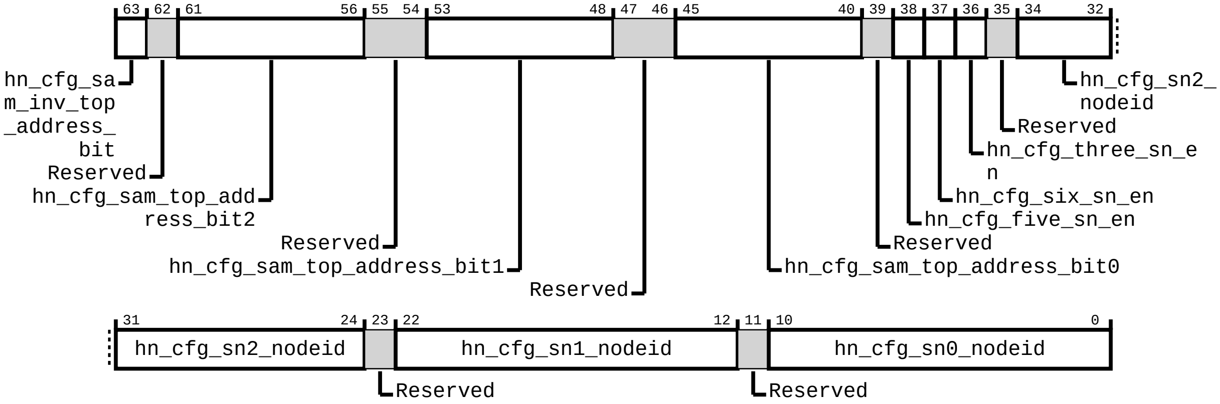


Table 5-466: por_hnf_sam_control attributes

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 and 6-SN configurations.	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 and 6-SN configurations.	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

5.3.15.64 por_hnf_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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-451: por_hnf_sam_memregion0

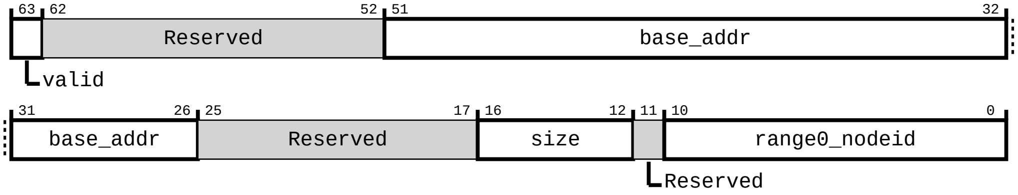


Table 5-467: por_hnf_sam_memregion0 attributes

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	-
[51:26]	base_addr	Base address of memory region 0 CONSTRAINT: Must be an integer multiple of region size.	RW	26'h0
[25:17]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[16:12]	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	5'h0
[11]	Reserved	Reserved	RO	-
[10:0]	range0_nodeid	Memory region 0 target node ID	RW	11'h0

5.3.15.65 por_hnf_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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-452: por_hnf_sam_memregion1

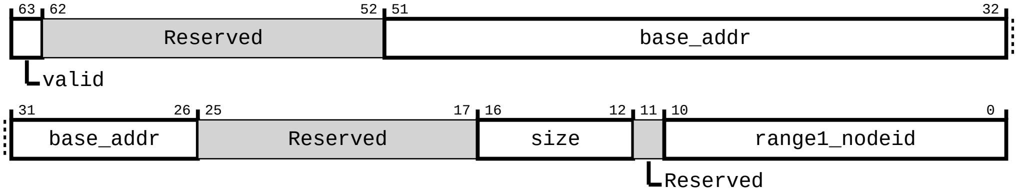


Table 5-468: por_hnf_sam_memregion1 attributes

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	-
[51:26]	base_addr	Base address of memory region 1 CONSTRAINT: Must be an integer multiple of region size.	RW	26'h0
[25:17]	Reserved	Reserved	RO	-
[16:12]	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	5'h0
[11]	Reserved	Reserved	RO	-
[10:0]	range1_nodeid	Memory region 1 target node ID	RW	11'h0

5.3.15.66 por_hnf_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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-453: por_hnf_sam_sn_properties

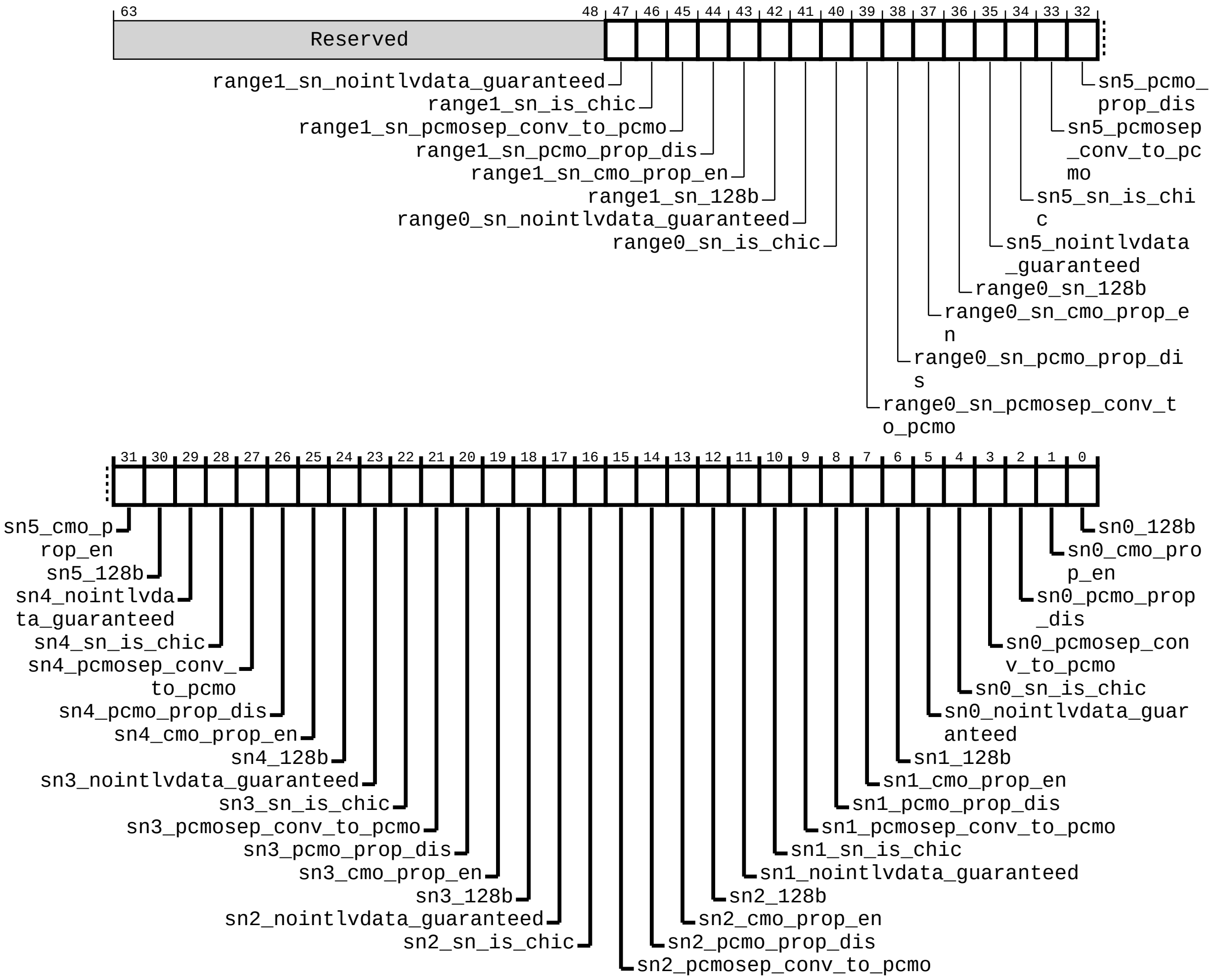


Table 5-469: por_hnf_sam_sn_properties attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47]	range1_sn_nointlvdata_guaranteed	SN guarantees that the return data will not be interleaved	RW	1'b0
[46]	range1_sn_is_chic	If set, indicates that the range 1 SN is a CHI-C SN	RW	1'b0
[45]	range1_sn_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for range 1 SN. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[44]	range1_sn_pcmo_prop_dis	If set, disables Persistent CMO (PCMO) propagation for range 1 SN	RW	1'b0
[43]	range1_sn_cmo_prop_en	Enables CMO propagation for range 1 SN	RW	1'b0

Bits	Name	Description	Type	Reset
[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 guarantees the return data is not interleaved	RW	1'b0
[40]	range0_sn_is_chic	If set, indicates that the range 0 SN is a CHI-C SN	RW	1'b0
[39]	range0_sn_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for range 0 SN. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[38]	range0_sn_pcmo_prop_dis	If set, disables Persistent CMO (PCMO) propagation for range 0 SN	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 guarantees the return data is not interleaved	RW	1'b0
[34]	sn5_sn_is_chic	If set, indicates that SN 5 is a CHI-C SN	RW	1'b0
[33]	sn5_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for SN 5. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[32]	sn5_pcmo_prop_dis	If set, disables PCMO propagation for SN 5	RW	1'b0
[31]	sn5_cmo_prop_en	If set, enables CMO propagation for SN 5	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 guarantees the return data is not interleaved	RW	1'b0
[28]	sn4_sn_is_chic	If set, indicates that SN 4 is a CHI-C SN	RW	1'b0
[27]	sn4_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for SN 4. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[26]	sn4_pcmo_prop_dis	If set, disables PCMO propagation for SN 4	RW	1'b0
[25]	sn4_cmo_prop_en	If set, enables CMO propagation for SN 4	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 guarantees the return data is not interleaved	RW	1'b0
[22]	sn3_sn_is_chic	If set, indicates that SN 3 is a CHI-C SN	RW	1'b0
[21]	sn3_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for SN 3. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[20]	sn3_pcmo_prop_dis	If set, disables PCMO propagation for SN 3	RW	1'b0

Bits	Name	Description	Type	Reset
[19]	sn3_cmo_prop_en	If set, enables CMO propagation for SN 3	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 guarantees the return data is not interleaved	RW	1'b0
[16]	sn2_sn_is_chic	If set, indicates that SN 2 is a CHI-C SN	RW	1'b0
[15]	sn2_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for SN 2. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[14]	sn2_pcmo_prop_dis	If set, disables PCMO propagation for SN 2	RW	1'b0
[13]	sn2_cmo_prop_en	If set, enables CMO propagation for SN 2	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 guarantees the return data is not interleaved	RW	1'b0
[10]	sn1_sn_is_chic	If set, indicates that SN 1 is a CHI-C SN	RW	1'b0
[9]	sn1_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for SN 1. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[8]	sn1_pcmo_prop_dis	If set, disables PCMO propagation for SN 1	RW	1'b0
[7]	sn1_cmo_prop_en	If set, enables CMO propagation for SN 1	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 guarantees the return data is not interleaved	RW	1'b0
[4]	sn0_sn_is_chic	If set, indicates that SN 0 is a CHI-C SN	RW	1'b0
[3]	sn0_pcmosep_conv_to_pcmo	If set, CleanSharedPersistSep operations are converted to CleanSharedPersist for SN 0. CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1.	RW	1'b0
[2]	sn0_pcmo_prop_dis	If set, disables PCMO propagation for SN 0	RW	1'b0
[1]	sn0_cmo_prop_en	If set, enables CMO propagation for SN 0	RW	1'b0
[0]	sn0_128b	Data width of SN 0: 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0

5.3.15.67 por_hnf_sam_6sn_nodeid

Configures node IDs for slave nodes 3-5 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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-454: por_hnf_sam_6sn_nodeid

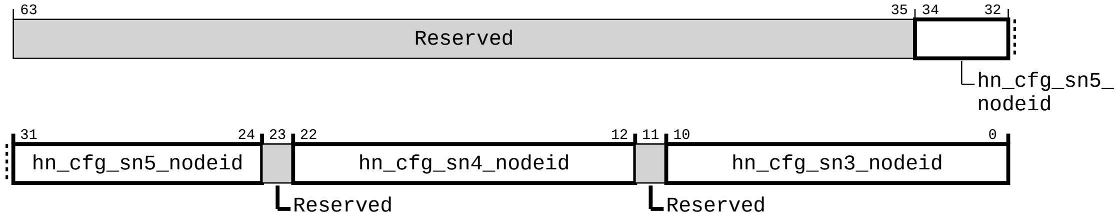


Table 5-470: por_hnf_sam_6sn_nodeid attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[34:24]	hn_cfg_sn5_nodeid	SN 5 node ID	RW	11'h0
[23]	Reserved	Reserved	RO	-
[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

5.3.15.68 por_hnf_sam_sn_properties1

Configures other 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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-455: por_hnf_sam_sn_properties1

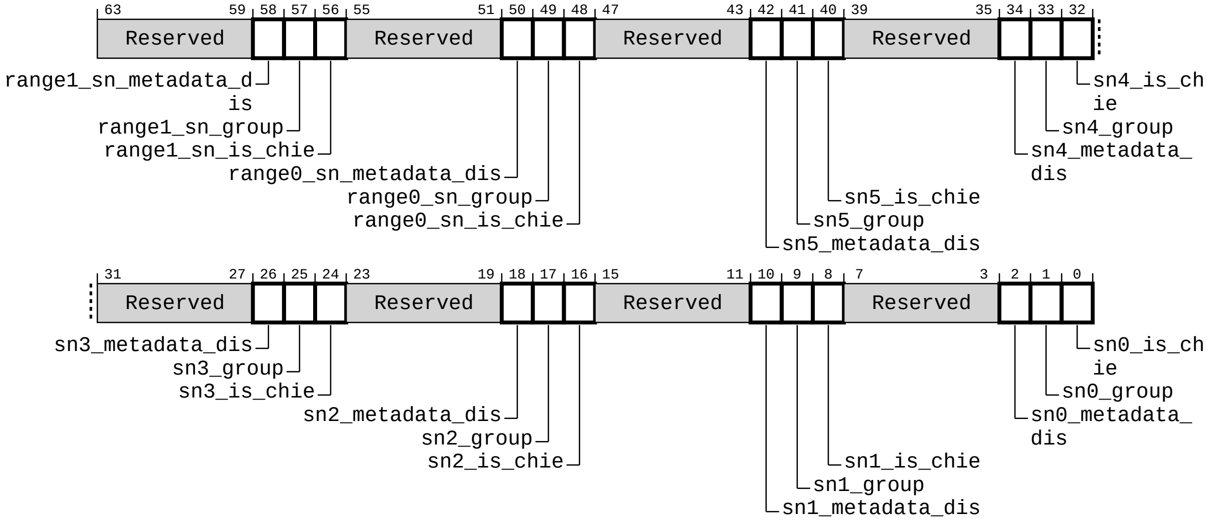


Table 5-471: por_hnf_sam_sn_properties1 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[58]	range1_sn_metadata_dis	If set, HN-F implements metadata termination flow for range 1 SN	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 (Not applicable in CMN-650)	RW	1'b0
[55:51]	Reserved	Reserved	RO	-
[50]	range0_sn_metadata_dis	If set, HN-F implements metadata termination flow for range 0 SN	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 (Not applicable in CMN-650)	RW	1'b0
[47:43]	Reserved	Reserved	RO	-
[42]	sn5_metadata_dis	If set, HN-F implements metadata termination flow for SN 5	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 (Not applicable in CMN-650)	RW	1'b0
[39:35]	Reserved	Reserved	RO	-
[34]	sn4_metadata_dis	If set, HN-F implements metadata termination flow for SN 4	RW	1'b0
[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 (Not applicable in CMN-650)	RW	1'b0
[31:27]	Reserved	Reserved	RO	-
[26]	sn3_metadata_dis	If set, HN-F implements metadata termination flow for SN 3	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 (Not applicable in CMN-650)	RW	1'b0
[23:19]	Reserved	Reserved	RO	-
[18]	sn2_metadata_dis	If set, HN-F implements metadata termination flow for SN 2	RW	1'b0
[17]	sn2_group	Specifies the SN-F grouping: 1'b0: Group A 1'b1: Group B	RW	1'b0

Bits	Name	Description	Type	Reset
[16]	sn2_is_chie	SN 2 supports CHI-E (Not applicable in CMN-650)	RW	1'b0
[15:11]	Reserved	Reserved	RO	-
[10]	sn1_metadata_dis	If set, HN-F implements metadata termination flow for SN 1	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 (Not applicable in CMN-650)	RW	1'b0
[7:3]	Reserved	Reserved	RO	-
[2]	sn0_metadata_dis	If set, HN-F implements metadata termination flow for SN 0	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 (Not applicable in CMN-650)	RW	1'b0

5.3.15.69 por_hnf_cml_port_aggr_grp0_add_mask

Configures the address mask for CCIX Port Aggregation Group (CPAG) 0.

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_hnf_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

Figure 5-456: por_hnf_cml_port_aggr_grp0_add_mask

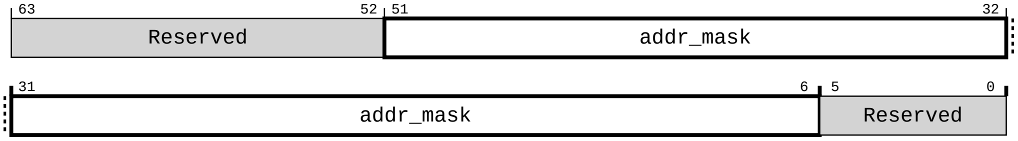


Table 5-472: por_hnf_cml_port_aggr_grp0_add_mask attributes

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	-

5.3.15.70 por_hnf_cml_port_aggr_grp1_add_mask

Configures the address mask for CCIX Port Aggregation Group (CPAG) 1.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF88

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_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

Figure 5-457: por_hnf_cml_port_aggr_grp1_add_mask

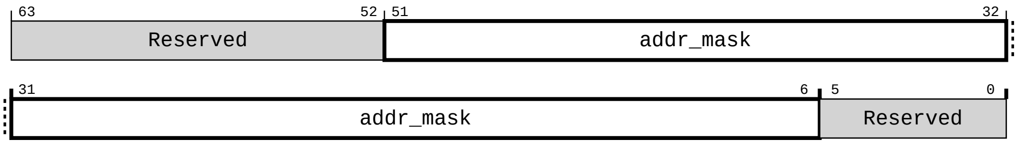


Table 5-473: por_hnf_cml_port_aggr_grp1_add_mask attributes

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	-

5.3.15.71 por_hnf_cml_port_aggr_grp2_add_mask

Configures the address mask for CCIX Port Aggregation Group (CPAG) 2.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF90

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_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

Figure 5-458: por_hnf_cml_port_aggr_grp2_add_mask

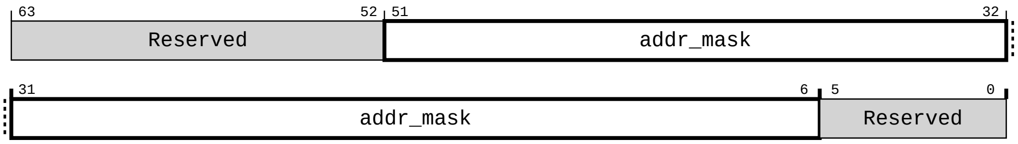


Table 5-474: por_hnf_cml_port_aggr_grp2_add_mask attributes

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	-

5.3.15.72 por_hnf_cml_port_aggr_grp3_add_mask

Configures the address mask for CCIX Port Aggregation Group (CPAG) 3.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF98

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_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

Figure 5-459: por_hnf_cml_port_aggr_grp3_add_mask

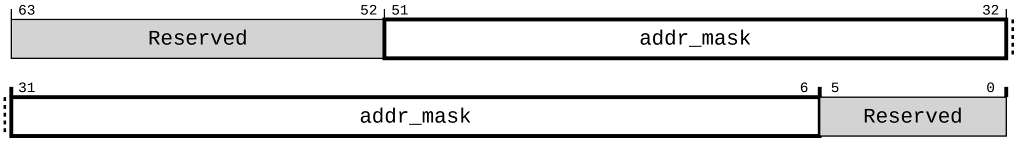


Table 5-475: por_hnf_cml_port_aggr_grp3_add_mask attributes

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	-

5.3.15.73 por_hnf_cml_port_aggr_grp4_add_mask

Configures the address mask for CCIX Port Aggregation Group (CPAG) 4.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hFA0

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_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

Figure 5-460: por_hnf_cml_port_aggr_grp4_add_mask

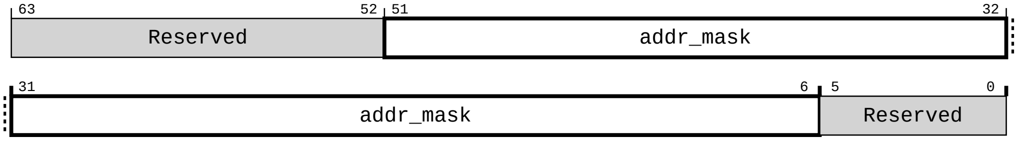


Table 5-476: por_hnf_cml_port_aggr_grp4_add_mask attributes

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	-

5.3.15.74 por_hnf_cml_port_aggr_grp_reg0

Configures the CCIX Port Aggregation Group (CPAG) port NodeIDs.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hFB0

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_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

Figure 5-461: por_hnf_cml_port_aggr_grp_reg0

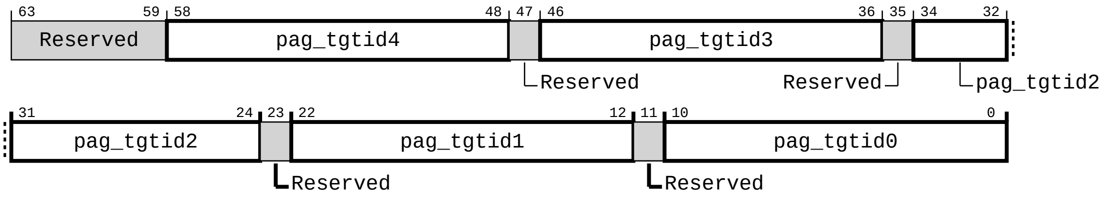


Table 5-477: por_hnf_cml_port_aggr_grp_reg0 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	pag_tgtid4	Specifies target ID 4 for CPAG	RW	11'b0
[47]	Reserved	Reserved	RO	-
[46:36]	pag_tgtid3	Specifies target ID 3 for CPAG	RW	11'b0
[35]	Reserved	Reserved	RO	-
[34:24]	pag_tgtid2	Specifies target ID 2 for CPAG	RW	11'b0
[23]	Reserved	Reserved	RO	-
[22:12]	pag_tgtid1	Specifies target ID 1 for CPAG	RW	11'b0
[11]	Reserved	Reserved	RO	-
[10:0]	pag_tgtid0	Specifies target ID 0 for CPAG	RW	11'b0

5.3.15.75 por_hnf_cml_port_aggr_grp_reg1

Configures the CCIX Port Aggregation Group (CPAG) port NodeIDs.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hFB8

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_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

Figure 5-462: por_hnf_cml_port_aggr_grp_reg1

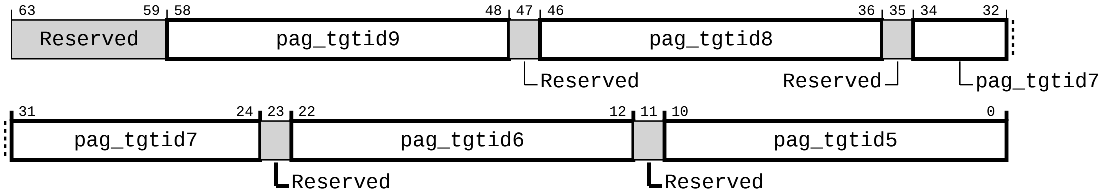


Table 5-478: por_hnf_cml_port_aggr_grp_reg1 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	pag_tgtid9	Specifies target ID 9 for CPAG	RW	11'b0
[47]	Reserved	Reserved	RO	-
[46:36]	pag_tgtid8	Specifies target ID 8 for CPAG	RW	11'b0
[35]	Reserved	Reserved	RO	-
[34:24]	pag_tgtid7	Specifies target ID 7 for CPAG	RW	11'b0
[23]	Reserved	Reserved	RO	-
[22:12]	pag_tgtid6	Specifies target ID 6 for CPAG	RW	11'b0
[11]	Reserved	Reserved	RO	-
[10:0]	pag_tgtid5	Specifies target ID 5 for CPAG	RW	11'b0

5.3.15.76 por_hnf_cml_port_aggr_ctrl_reg

Configures the CCIX Port Aggregation Groups (CPAGs).

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

por_hnf_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

Figure 5-463: por_hnf_cml_port_aggr_ctrl_reg

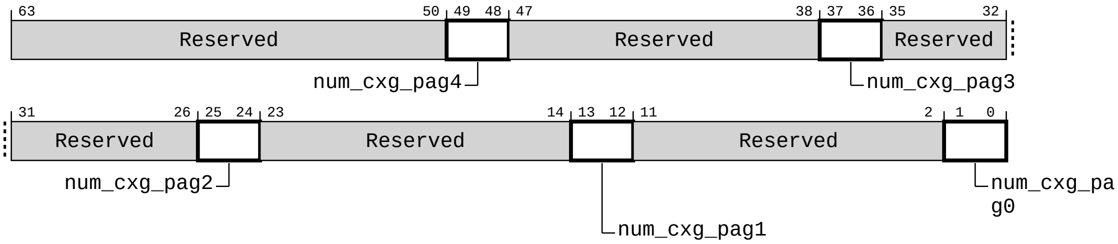


Table 5-479: por_hnf_cml_port_aggr_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:50]	Reserved	Reserved	RO	-
[49:48]	num_cxg_pag4	Specifies the number of CXRAs in CPAG 4. CONSTRAINT: Can use pag_tgtid8 through pag_tgtid9 of por_hnf_cml_port_aggr_grp_reg1. 2'b00: One port used 2'b01: Two ports used 2'b10: Reserved 2'b11: Reserved	RW	2'b0
[47:38]	Reserved	Reserved	RO	-
[37:36]	num_cxg_pag3	Specifies the number of CXRAs in CPAG 3. CONSTRAINT: Can use pag_tgtid6 through pag_tgtid7 of por_hnf_cml_port_aggr_grp_reg1. 2'b00: One port used 2'b01: Two ports used 2'b10: Reserved 2'b11: Reserved	RW	2'b0
[35:26]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[25:24]	num_cxg_pag2	Specifies the number of CXRAs in CPAG 2. CONSTRAINT: Can use pag_tgtid4 through pag_tgtid7 of por_hnf_cml_port_aggr_grp_reg[0,1]. 2'b00: One port used 2'b01: Two ports used 2'b10: Four ports used 2'b11: Reserved	RW	2'b0
[23:14]	Reserved	Reserved	RO	-
[13:12]	num_cxg_pag1	Specifies the number of CXRAs in CPAG 1. CONSTRAINT: Can use pag_tgtid2 through pag_tgtid3 of por_hnf_cml_port_aggr_grp_reg0. 2'b00: One port used 2'b01: Two ports used 2'b10: Reserved 2'b11: Reserved	RW	2'b0
[11:2]	Reserved	Reserved	RO	-
[1:0]	num_cxg_pag0	Specifies the number of CXRAs in CPAG 0. CONSTRAINT: Can use pag_tgtid0 through pag_tgtid7 of por_hnf_cml_port_aggr_grp_reg[0,1]. 2'b00: One port used 2'b01: Two ports used 2'b10: Four ports used 2'b11: Eight ports used	RW	2'b0

5.3.15.77 por_hnf_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

por_hnf_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

Figure 5-464: por_hnf_abf_lo_addr

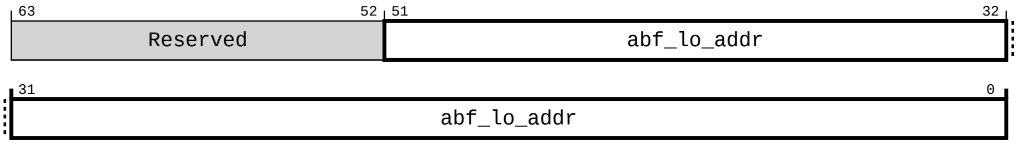


Table 5-480: por_hnf_abf_lo_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:0]	abf_lo_addr	Lower address range for ABF	RW	52'b0

5.3.15.78 por_hnf_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

por_hnf_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

Figure 5-465: por_hnf_abf_hi_addr

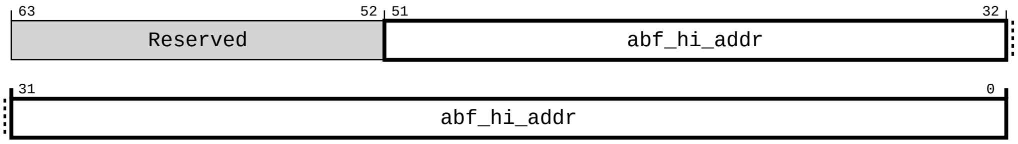


Table 5-481: por_hnf_abf_hi_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:0]	abf_hi_addr	Upper address range for ABF	RW	52'b0

5.3.15.79 por_hnf_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

por_hnf_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

Figure 5-466: por_hnf_abf_pr

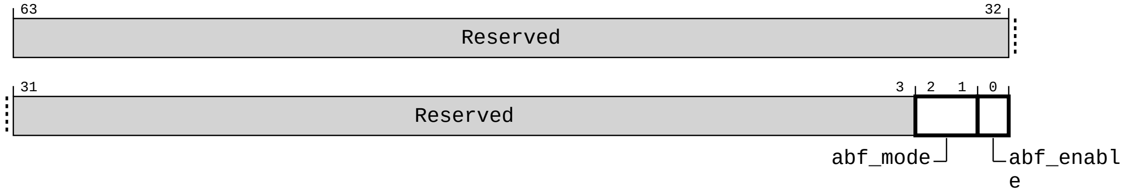


Table 5-482: por_hnf_abf_pr attributes

Bits	Name	Description	Type	Reset
[63:3]	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 ABF based on high and low address ranges	RW	1'b0

5.3.15.80 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-467: por_hnf_abf_sr

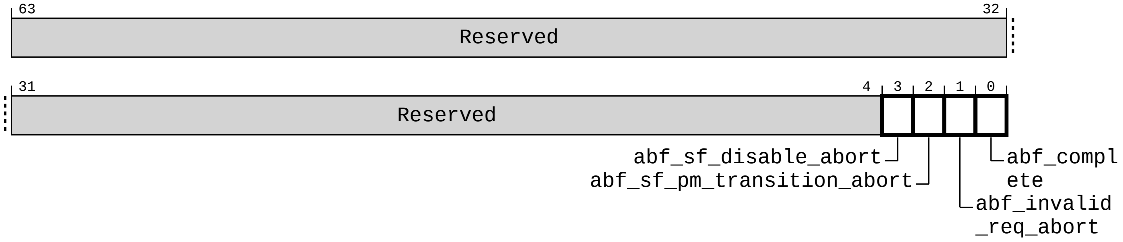


Table 5-483: por_hnf_abf_sr attributes

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, or SF_ONLY mode. In this case, request is aborted.	RO	1'b0
[0]	abf_complete	ABF completed	RO	1'b0

5.3.15.81 por_hnf_cbusy_write_limit_ctl

Completer Busy (CBusy) threshold limits for POCQ write entries. CONSTRAINT: hnf_adv_cbusy_mode_dis must be 1'b0 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

Figure 5-468: por_hnf_cbusy_write_limit_ctl

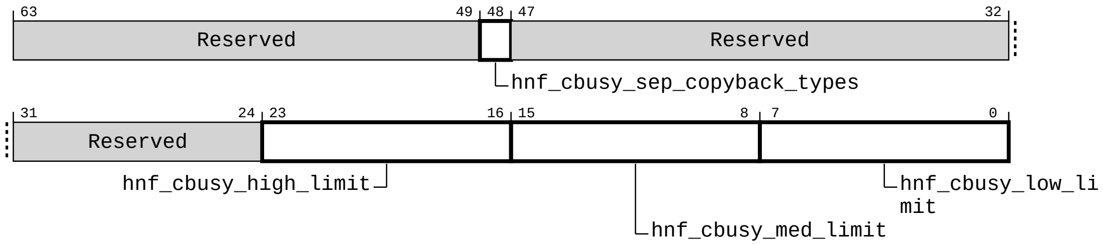


Table 5-484: por_hnf_cbusy_write_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:49]	Reserved	Reserved	RO	-
[48]	hnf_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]	hnf_cbusy_high_limit	POCQ limit for write CBusy High	RW	Configuration dependent
[15:8]	hnf_cbusy_med_limit	POCQ limit for write CBusy Medium	RW	Configuration dependent
[7:0]	hnf_cbusy_low_limit	POCQ limit for write CBusy Low	RW	Configuration dependent

5.3.15.82 por_hnf_cbusy_resp_ctl

Controls the responses sent from HN-F to RN-F. CONSTRAINT: hnf_adv_cbusy_mode_dis must be 1'b0 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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-469: por_hnf_cbusy_resp_ctl

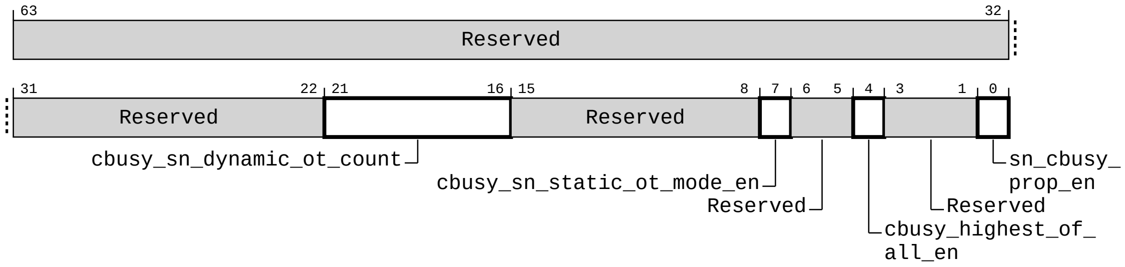


Table 5-485: por_hnf_cbusy_resp_ctl attributes

Bits	Name	Description	Type	Reset
[63:22]	Reserved	Reserved	RO	-
[21:16]	cbusy_sn_dynamic_ot_count	Specifies the granularity at which HN-F dynamically throttles transactions to SN-F. CONSTRAINT: The permitted values are 2, 4, or 8.	RW	6'b000100
[15:8]	Reserved	Reserved	RO	-
[7]	cbusy_sn_static_ot_mode_en	Controls CBusy between HN-F and SN-F: 1'b0: HN-F dynamically throttles outstanding requests to SN-F. 1'b1: HN-F uses fixed transactions count at each CBusy level at 1/4th POCQ granularity.	RW	1'b0
[6:5]	Reserved	Reserved	RO	-
[4]	cbusy_highest_of_all_en	Controls CBusy between HN-F and SN-F: 1'b0: Sends the HN-F or SN-F as configured 1'b1: Selects 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: CBusy of HN-F POCQ is sent 1'b1: CBusy of SN-F is sent	RW	1'b0

5.3.15.83 por_hnf_cbusy_sn_ctl

Controls the SN-F CBusy thresholds. CONSTRAINT: hnf_adv_cbusy_mode_dis must be 1'b0 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

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-470: por_hnf_cbusy_sn_ctl

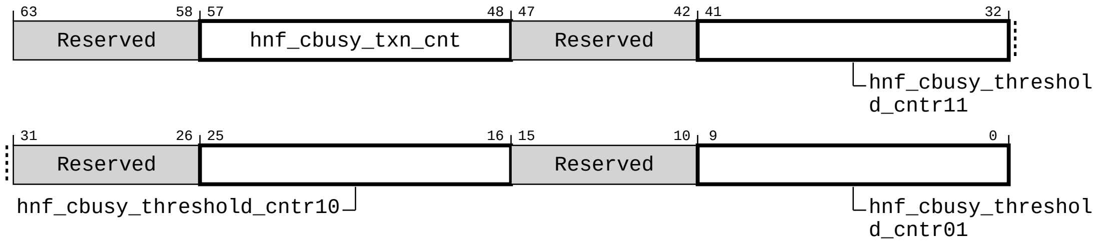


Table 5-486: por_hnf_cbusy_sn_ctl attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	hnf_cbusy_txn_cnt	Number of transactions over which the counters are tracked	RW	10'b0100000000
[47:42]	Reserved	Reserved	RO	-
[41:32]	hnf_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]	hnf_cbusy_threshold_cntr10	CBusy threshold at which SN-F is considered busy for Counter_10	RW	10'b0000100000
[15:10]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[9:0]	hnf_cbusy_threshold_cntr01	CBusy threshold at which SN-F is considered busy for Counter_01	RW	10'b0001000000

5.3.15.84 por_hnf_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

por_hnf_secure_register_groups_override.partner_scratch_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-471: por_hnf_partner_scratch_reg0

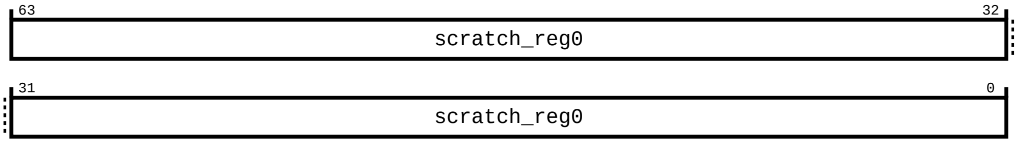


Table 5-487: por_hnf_partner_scratch_reg0 attributes

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg0	64-bit scratch register 0 with read/write access	RW	64'h00000000

5.3.15.85 por_hnf_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

por_hnf_secure_register_groups_override.partner_scratch_override

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-472: por_hnf_partner_scratch_reg1

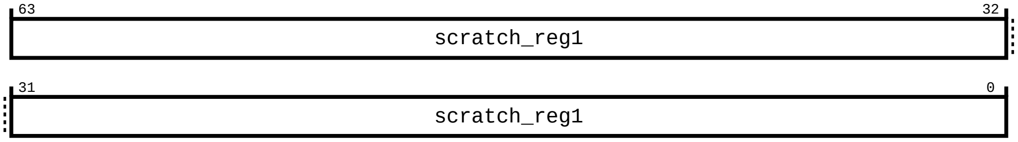


Table 5-488: por_hnf_partner_scratch_reg1 attributes

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg1	64-bit scratch register 1 with read/write access	RW	64'h00000000

5.3.15.86 por_hnf_cfg_slcsf_dbgrd

Controls access modes for SLC tag, 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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-473: por_hnf_cfg_slcsf_dbgrd

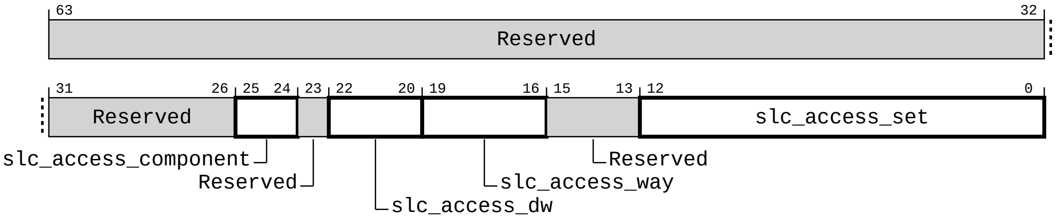


Table 5-489: por_hnf_cfg_slcsf_dbgrd attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:24]	slc_access_component	Specifies SLC and SF array debug read: 2'b01: SLC data read 2'b10: SLC tag read 2'b11: SF tag read	WO	2'b00
[23]	Reserved	Reserved	RO	-
[22:20]	slc_access_dw	64-bit chunk address for SLC data debug read access	WO	3'h0
[19:16]	slc_access_way	Way address for SLC and SF debug read access	WO	4'h0
[15:13]	Reserved	Reserved	RO	-
[12:0]	slc_access_set	Set address for SLC and SF debug read access	WO	13'h0

5.3.15.87 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-474: por_hnf_slc_cache_access_slc_tag

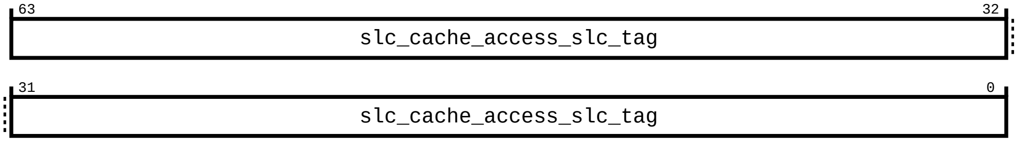


Table 5-490: por_hnf_slc_cache_access_slc_tag attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_tag	SLC tag debug read data	RO	64'h0

5.3.15.88 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-475: por_hnf_slc_cache_access_slc_tag1

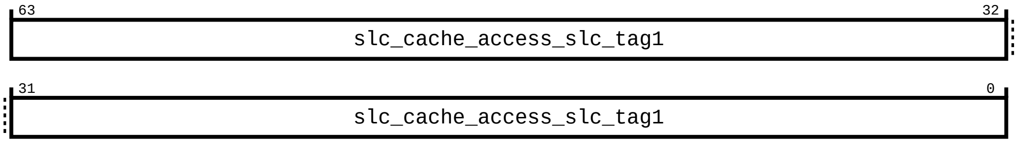


Table 5-491: por_hnf_slc_cache_access_slc_tag1 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_tag1	SLC tag debug read data	RO	64'h0

5.3.15.89 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-476: por_hnf_slc_cache_access_slc_data

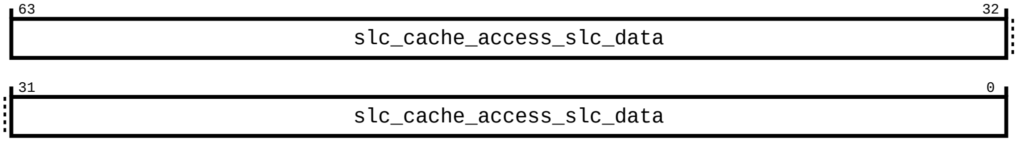


Table 5-492: por_hnf_slc_cache_access_slc_data attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_data	SLC data RAM debug read data	RO	64'h0

5.3.15.90 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-477: por_hnf_slc_cache_access_slc_mte_tag

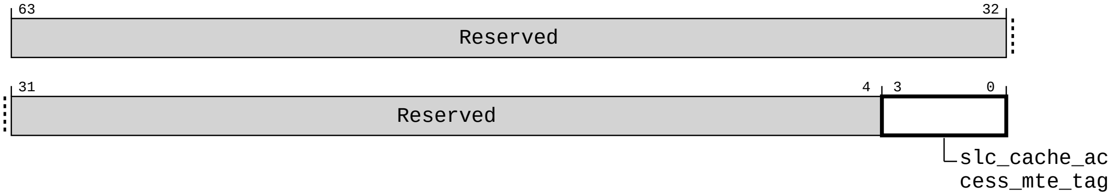


Table 5-493: por_hnf_slc_cache_access_slc_mte_tag attributes

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 (128-bit chunk of data)	RO	4'h0

5.3.15.91 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-478: por_hnf_slc_cache_access_sf_tag

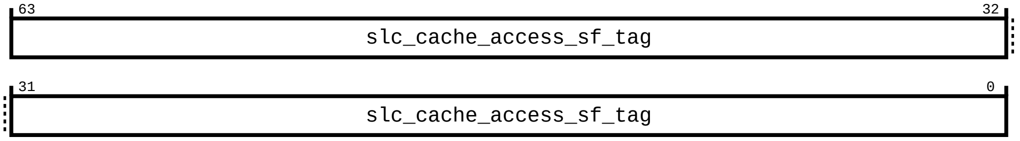


Table 5-494: por_hnf_slc_cache_access_sf_tag attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag	SF tag debug read data	RO	64'h0

5.3.15.92 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-479: por_hnf_slc_cache_access_sf_tag1

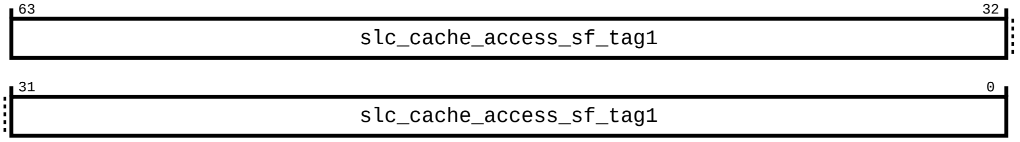


Table 5-495: por_hnf_slc_cache_access_sf_tag1 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag1	SF tag debug read data	RO	64'h0

5.3.15.93 por_hnf_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

por_hnf_secure_register_groups_override.slcsf_dbgrd

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-480: por_hnf_slc_cache_access_sf_tag2

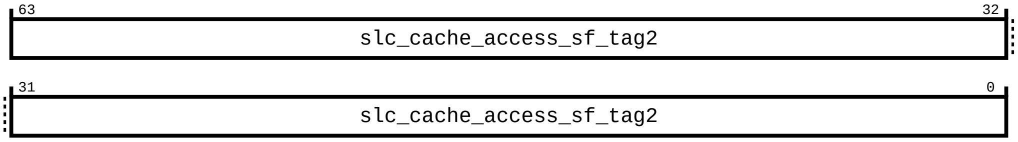


Table 5-496: por_hnf_slc_cache_access_sf_tag2 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag2	SF tag debug read data	RO	64'h0

5.3.15.94 por_hnf_pmu_event_sel

Specifies the Performance Monitoring Unit (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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-481: por_hnf_pmu_event_sel

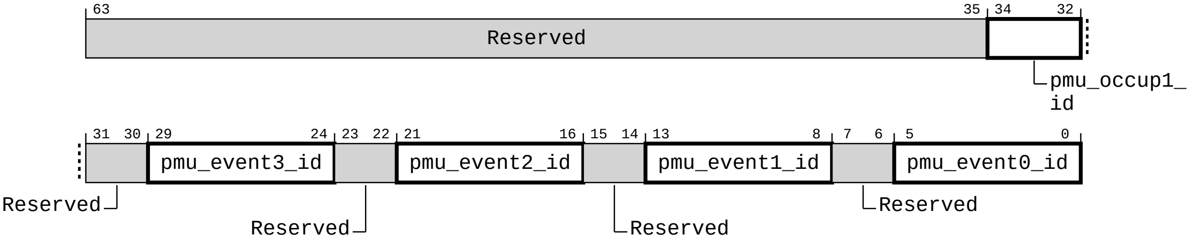


Table 5-497: por_hnf_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:35]	Reserved	Reserved	RO	-
[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	RW	3'h0

Bits	Name	Description	Type	Reset
[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) 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	RW	6'h00

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	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, and 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 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	RW	6'h00

5.3.15.95 por_hnf_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

Only accessible by Secure accesses.

Bit descriptions

Figure 5-482: por_hnf_pmu_mpam_sel

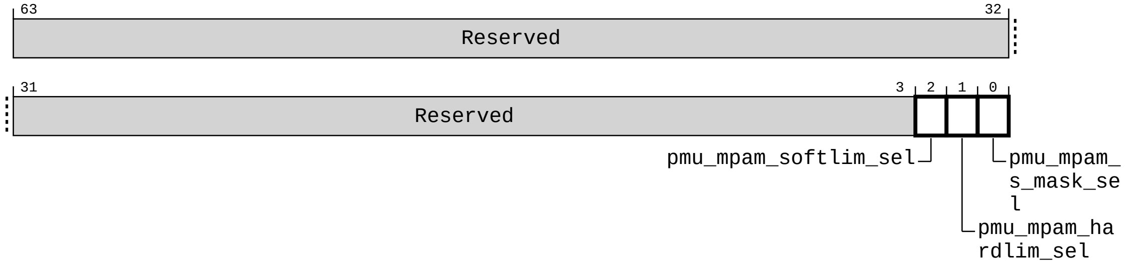


Table 5-498: por_hnf_pmu_mpam_sel attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	pmu_mpam_softlim_sel	When set, HN-F Performance Monitoring Unit (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 that are indicated in filter register	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	pmu_mpam_hardlim_sel	If 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	If 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

5.3.15.96 por_hnf_pmu_mpam_pardid_mask0-7

There are 7 iterations of this register, parameterized by the index from 0 to 7. Functions as mask for PARTID[#{64*(index+1)-1}:#{64*index}] filter for MPAM PMU events

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h2010 + (8 × #[0, 1, ... 7])

Type

RW

Reset value

See individual bit resets

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-483: por_hnf_pmu_mpam_pardid_mask0-7

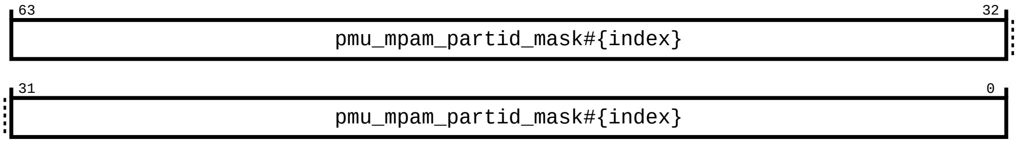


Table 5-499: por_hnf_pmu_mpam_pardid_mask0-7 attributes

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 Note: This mask is used only when por_hnf_pmu_mpam_sel is set for PARTID based counting.	RW	64'b0

5.3.15.97 por_hnf_rn_cluster0-63_physid_reg0

There are 64 iterations of this register, parameterized by the index 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 + (32 × #[0, 1, ... 63])

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-484: por_hnf_rn_cluster0-63_physid_reg0

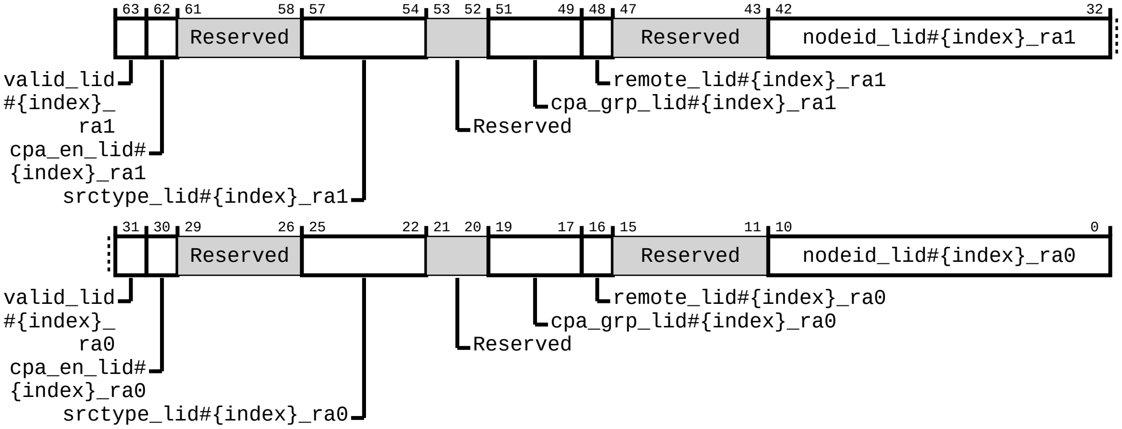


Table 5-500: por_hnf_rn_cluster0-63_physid_reg0 attributes

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 CPA 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 Others : Reserved	RW	4'h0000
[53:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:49]	cpa_grp_lid#{index}_ra1	Specifies CPAG ID 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[21:20]	Reserved	Reserved	RO	-
[19:17]	cpa_grp_lid#{index}_ra0	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0

Bits	Name	Description	Type	Reset
[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

5.3.15.98 por_hnf_rn_cluster64-127_physid_reg0

There are 64 iterations of this register, parameterized by the index 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 + (32 × #[64, 65, ... 127])

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-485: por_hnf_rn_cluster64-127_physid_reg0

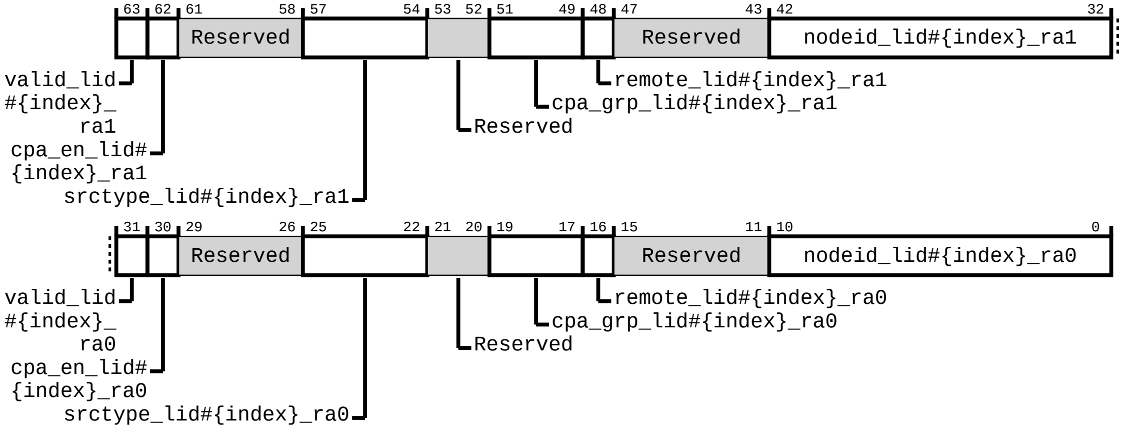


Table 5-501: por_hnf_rn_cluster64-127_physid_reg0 attributes

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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[53:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:49]	cpa_grp_lid#{index}_ra1	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[21:20]	Reserved	Reserved	RO	-
[19:17]	cpa_grp_lid#{index}_ra0	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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

Bits	Name	Description	Type	Reset
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra0	Specifies the node ID	RW	11'h0

5.3.15.99 por_hnf_rn_cluster0-127_physid_reg1

There are 128 iterations of this register, parameterized by the index 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'h3C00 + (32 × #[0, 1, ... 127])

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-486: por_hnf_rn_cluster0-127_physid_reg1

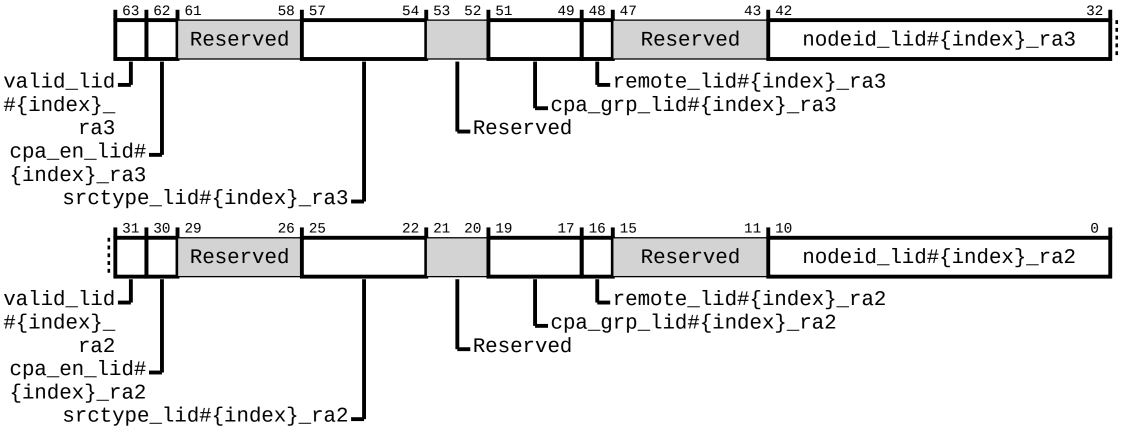


Table 5-502: por_hnf_rn_cluster0-127_physid_reg1 attributes

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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[53:52]	Reserved	Reserved	RO	-
[51:49]	cpa_grp_lid#{index}_ra3	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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 CCIX Port Aggregation (CPA) is enabled: 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0

Bits	Name	Description	Type	Reset
[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 Others : Reserved	RW	4'h0000
[21:20]	Reserved	Reserved	RO	-
[19:17]	cpa_grp_lid#{index}_ra2	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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

5.3.15.100 por_hnf_rn_cluster0-127_physid_reg2

There are 128 iterations of this register, parameterized by the index 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'h3C00 + (32 × #[0, 1, ... 127])

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-487: por_hnf_rn_cluster0-127_physid_reg2

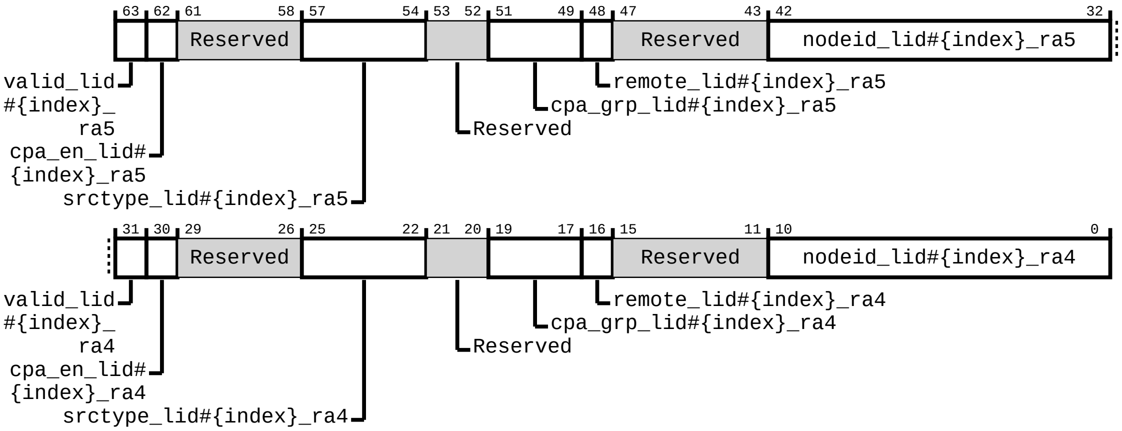


Table 5-503: por_hnf_rn_cluster0-127_physid_reg2 attributes

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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[53:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:49]	cpa_grp_lid#{index}_ra5	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[21:20]	Reserved	Reserved	RO	-
[19:17]	cpa_grp_lid#{index}_ra4	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0

Bits	Name	Description	Type	Reset
[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

5.3.15.101 por_hnf_rn_cluster0-127_physid_reg3

There are 128 iterations of this register, parameterized by the index 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'h3C00 + (32 × #[0, 1, ... 127])

Type

RW

Reset value

See individual bit resets

Secure group override

por_hnf_secure_register_groups_override.sam_control

Usage constraints

Only accessible by Secure accesses.

Bit descriptions

Figure 5-488: por_hnf_rn_cluster0-127_physid_reg3

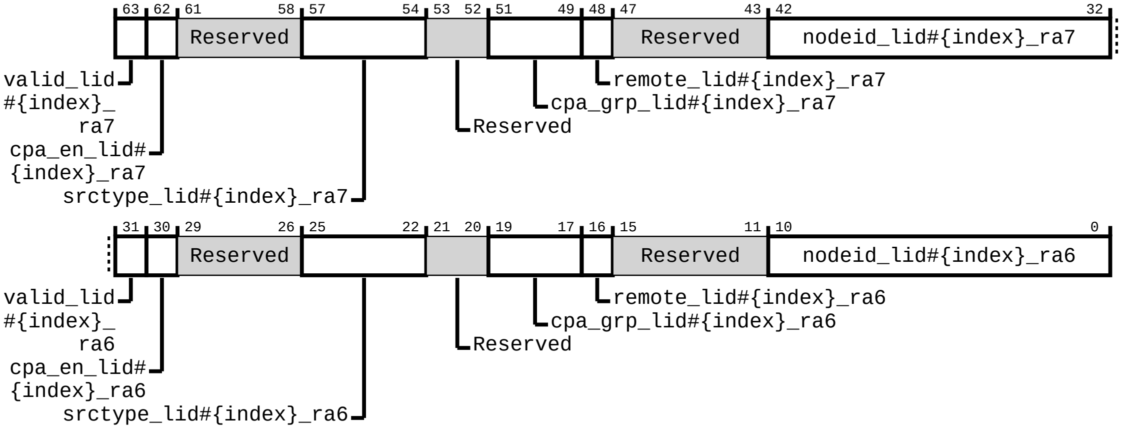


Table 5-504: por_hnf_rn_cluster0-127_physid_reg3 attributes

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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[53:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:49]	cpa_grp_lid#{index}_ra7	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'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 CCIX Port Aggregation (CPA) 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 Others : Reserved	RW	4'h0000
[21:20]	Reserved	Reserved	RO	-
[19:17]	cpa_grp_lid#{index}_ra6	Specifies CCIX Port Aggregation Group (CPAG) ID: 3'b000: CPAG ID 0 3'b001: CPAG ID 1 3'b010: CPAG ID 2 3'b011: CPAG ID 3 3'b100: CPAG ID 4	RW	3'h0

Bits	Name	Description	Type	Reset
[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

5.4 CMN-650 programming

This section contains CMN-650 programming information.

5.4.1 Boot-time programming sequence

A specific boot-time programming sequence must be used to set up CMN-650 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-650 components is available:

1. CMN-650 uses a default configuration to access boot flash through the HN-D ACE-Lite master interface and also the configuration registers.
2. An RN-F, or a master that is connected to an RN-I, must then access the configuration registers to configure CMN-650. This boot-time configuration must happen before there is broader access to components such as HN-F or SN.

The following example provides more information on the boot process. It assumes an SCP is performing the CMN-650 configuration.

1. The SCP boots, either from local memory or through CMN-650 memory accesses targeting memory behind the HN-D:
    - All other masters are either held in reset or issue no requests to CMN-650 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-650 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-650 configuration registers to program the SAM for all HN-Fs.
6. The SCP writes to the CMN-650 configuration registers to program the SAM for all RNs including the one being used by the SCP.

    Note

    RN-F ESAM interfaces are active and accept transactions before and during RN SAM programming. Therefore, transactions requiring RN SAM programming must be stalled or prevented until programming is complete.

    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.

Once the preceding steps are complete, the SCP can make general accesses anywhere in the address space and other masters can begin issuing requests.

5.4.2 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**, is 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. For more information, see 4.2.9 RN entry to and exit from snoop and DVM domains on page 85.

5.4.3 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.

5.4.3.1 Program the SAM

The SAM must be programmed using a specific sequence. An RN-F or master that is connected to an RN-I must perform this sequence during the configuration of CMN-650 at boot.

Before you begin

This sequence is part of the overall CMN-650 boot configuration process. There are steps that must occur at boot-time before SAM programming. For more information about the full process, see 5.4.1 Boot-time programming sequence on page 837.
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. For more information, see the following sections:

- 4.5 RN SAM on page 96
- 4.7 HN-F SAM on page 116

Procedure

1. Define the following memory map regions:
    - Hashed memory regions, which target HN-Fs. The hashed memory regions can be partitioned into SCGs, if applicable.
    - Non-hashed memory regions, which likely target HN-I, HN-D, or HN-P.
    - 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.

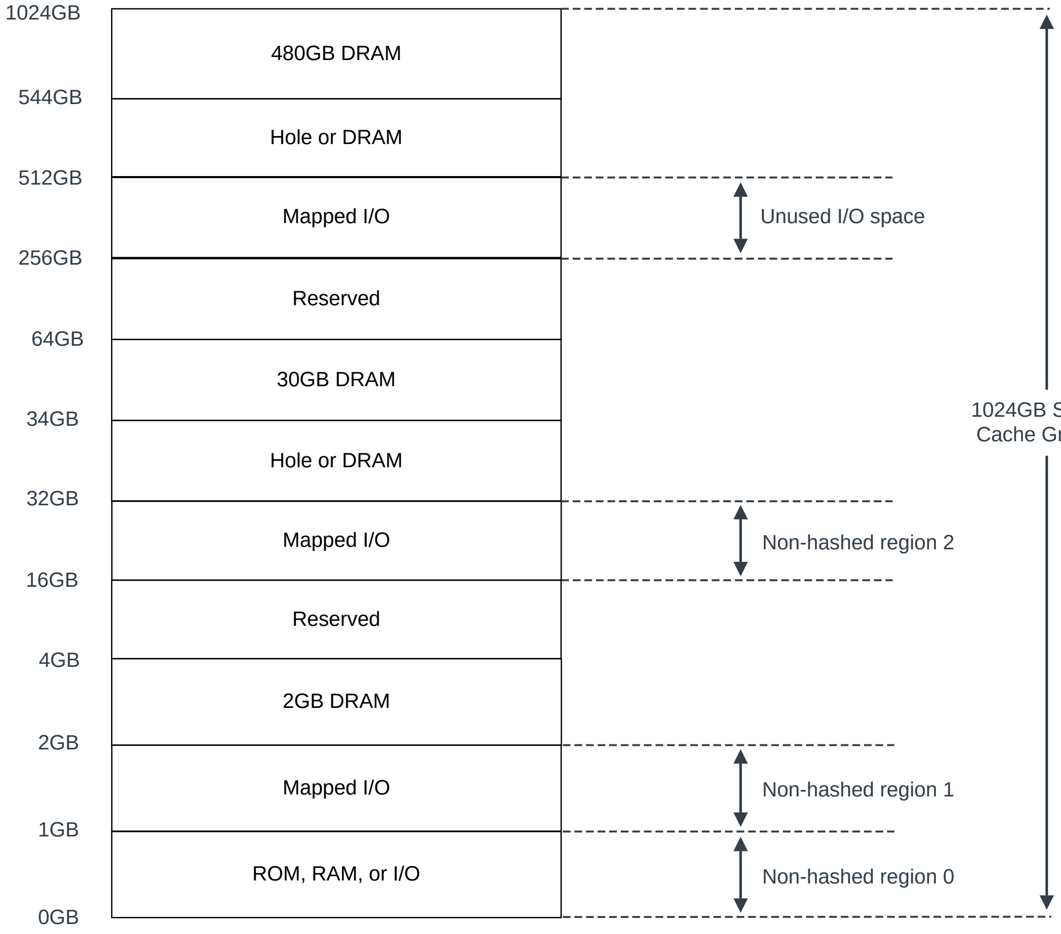
3. Program the following attributes and registers for each HN-F SAM:
    a) Program the appropriate properties for each SN-F ID, according to the features that are supported in the por_hnf_sam_sn_properties register.
       These properties provide the interface width as either 128-bit or 256-bit, CMO 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 PrefetchTgt 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.

5.4.3.2 Example memory map programming

This section describes an example memory map and how to program it in the RN SAM and HN-F SAM.

The following figure shows an example memory map with 1024GB addressable size. It is based on the Arm 40-bit proposed address map. It has three separate DRAM regions (in the address ranges from 2-4GB, 34-64GB and 544-1024GB) and four I/O regions, which must be mapped to specific targets. It is assumed that the I/O region 256-512GB is unused and no requests are sent to this address.

Figure 5-489: CMN-650 example memory map



It is assumed that there are eight HN-Fs in the system and all HN-Fs are being used for one SCG (group 0). To program the RN SAM, follow these steps:

1. Map the full 1024GB memory map to the system cache group. Arm recommends this mapping because DRAM regions are non-contiguous and the entire DRAM space is assigned to one SCG.

2. Carve out each of the non-hashed regions from the full 1024GB memory map as shown in the preceding figure. Assign each non-hashed region to individual non-hashed targets.
3. When the RN SAM programming is done, turn ON the region-based target ID selection by disabling the default mode of the RN SAM.

The following table shows the RN SAM registers and the corresponding programmed values.

Table 5-505: RN SAM registers and programmed values

Register name	Field name	Value	Description
sys_cache_grp_region0	region0_base_address	0x0_0000	Base address [51:26]
	region0_size	7'b0001110	1024GB size
	region0_target_type	2'b00	HN-F target type
	region0_valid	1'b1	Region 0 is valid.
sys_cache_grp_hn_nodeid_reg0	nodeid_0	<hnf0_node_id>	Physical node IDs of the HN-Fs in the system from HN-F 0 to HN-F 7. If RN-Fs are generating PrefetchTarget operations, then you must program the SN node IDs corresponding to each HN-F in sys_cache_grp_sn_nodeid_regX registers.
	nodeid_1	<hnf1_node_id>	
	nodeid_2	<hnf2_node_id>	
	nodeid_3	<hnf3_node_id>	
sys_cache_grp_hn_nodeid_reg1	nodeid_4	<hnf4_node_id>	
	nodeid_5	<hnf5_node_id>	
	nodeid_6	<hnf6_node_id>	
	nodeid_7	<hnf7_node_id>	
sys_cache_group_hn_count	scg0_num_hnf	0x08	Total of eight HN-Fs in this system cache group.
non_hash_mem_region_reg0	region0_base_address	0x0_0000	1GB from [51:26] 0x0_0000_0000
	region0_size	7'b0000100	1GB size
	region0_target_type	2'b01	HN-I target type
	region0_valid	1'b1	Region 0 is valid.
non_hash_mem_region_reg1	region1_base_address	0x0_0010	1GB region from 0x0_4000_0000
	region1_size	7'b0000100	1GB size
	region1_target_type	2'b01	HN-I target type
	region1_valid	1'b1	Region 1 is valid.
non_hash_mem_region_reg2	region2_base_address	0x0_0100	16GB region from 0x4_0000_0000
	region2_size	7'b0001000	16GB size
	region2_target_type	2'b01	HN-I target type
	region2_valid	1'b1	Region 2 is valid.
non_hash_tgt_nodeid0	nodeid_0	<hni0_node_id>	Node ID of HN-I 0 corresponding to non-hashed region 0
	nodeid_1	<hni1_node_id>	Node ID of HN-I 1 corresponding to non-hashed region 1
	nodeid_2	<hni2_node_id>	Node ID of HN-I 2 corresponding to non-hashed region 2
rnsam_status	nstall_req	1'b1	Unstall any operations that depend on SAM programming.
	default_target	1'b0	Disable default mode and use the programmed ranges for new incoming addresses.

Similarly to RN SAM, HN-F SAM must also be programmed so that it can select the correct SN-F target ID. All HN-F SAMs within SCG 0 must have the same programming, as the following table shows, including the attributes of each SN-F.

Table 5-506: HN-F programming information

Register name	Field name	Value	Description
por_hnf_sam_control	hn_cfg_sn0_nodeid	<sn0_node_id>	Node ID of SN-F 0
	hn_cfg_sn1_nodeid	<sn1_node_id>	Node ID of SN-F 1
	hn_cfg_sn2_nodeid	<sn2_node_id>	Node ID of SN-F 2
	hn_cfg_three_sn_en	1'b1	Enable 3-SN mode.
	hn_cfg_sam_top_address_bit1	39	Bit 39 of address
	hn_cfg_sam_top_address_bit0	36	Bit 36 of address
	hn_cfg_sam_inv_top_address_bit	1'b1	Invert top address bit.

5.4.4 Program the dual DAT/RSP channel selection scheme

Boot-programmable registers control the dual DAT/RSP channel selection scheme by forming a TgtID LUT. Programming these registers at boot overrides the default channel selection scheme.

About this task

When the `2XDATRSP_EN` parameter is enabled, a default channel selection scheme is applied. For more information about the scheme, see 4.10.2 Dual DAT/RSP channel selection on page 140. This scheme is active until you configure the dual DAT/RSP 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 DAT/RSP registers. You must program these nodes to 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.

If your configuration uses CALs, then all the devices that are attached to the CAL must use the same DAT/RSP channel. Therefore, you must program the TgtID of both devices into the registers.

Procedure

1. Identify and classify the targets that must map to channel 0 and channel 1, for both DAT and RSP channels.
2. Program the TGTID and CHN_SEL fields in por_mxp_multi_dat_rsp_chn_sel_* registers with the TgtIDs to map to the required channel, 0 or 1.
3. Set the VALID bit in each por_mxp_multi_dat_rsp_chn_sel_* register to indicate that you have configured valid TgtIDs in each register.
4. Set the multi_dat_rsp_chn_sel_programmed bit in the por_mxp_multi_dat_rsp_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_dat_rsp_chn_sel_* registers for all MXPs.

5.4.5 Program non-XY routing registers

To configure the behavior of the CMN-650 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 on how to avoid deadlocks when setting up this feature, see 4.10.8 Rules for avoiding deadlocks in non-XY routing on page 148.

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.
- 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.

5.4.6 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 4.9 HN-I SAM on page 129.
If you map an HN-I SAM address region to a PCIe slave, you must map all address regions of that HN-I SAM to PCIe slaves.

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 masters are present upstream.
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 provide 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.

5.4.7 DT programming

You must follow several programming sequences to set up DTM watchpoints and DTC correctly.

For more information about the CMN-650 DT functionality, see 7 Debug trace and PMU on page 902.

5.4.7.1 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_sel field 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 or secondary 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.

5.4.7.2 Program DTC

Use this procedure to set up the CMN-650 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.

5.4.8 PMU system programming

You must follow specific programming sequences to set up the PMU, PMU snapshot, and PMU interrupt functionality correctly.

5.4.8.1 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 por_dtm_pmu_config register.

   Note

   To activate CXLA PMU function, program the en_cxla_pmucmd_prop field of the associated por_cxg_ra_cfg_ctl register to 1.

7. Write 1 to the dtm_enable field of the por_dtm_control register.
8. Program the cntcfg field of the por_dt_pmcr register to pair the 32-bit global counters to make a 64-bit counter.
9. Write 1 to the dt_en field of the por_dt_dtc_ctl register.
10. Write 1 to the ovfl_intr_en field of the por_dt_pmcr register to enable interrupts on **INTREQPMU** on any global counter overflow.
11. Write 1 to the pmu_en field of the por_dt_pmcr register to start PMU operation.

5.4.8.2 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 5.4.8.1 Set up PMU counters on page 847.
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.

5.4.8.3 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 5.4.8.1 Set up PMU counters on page 847.
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**.

5.5 CML programming

The system must be programmed to enable correct operation with CML.

5.5.1 CML-related programmable registers

This section contains a list of CML programmable registers.

CXRA

- RA SAM address region registers (por_cxg_ra_sam_addr_region_reg<X>)

- LDID to RAID LUT registers:
    - por_cxg_ra_rnf_ldid_to_exp_raid_reg<X>
    - por_cxg_ra_rni_ldid_to_exp_raid_reg<X>
    - por_cxg_ra_rnd_ldid_to_exp_raid_reg<X>
- RAID or HAID to LinkID LUT registers:
    - por_cxg_ra_agentid_to_linkid_reg<X>
    - por_cxg_ra_agentid_to_linkid_val
    - por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg_0-127
- CCIX Protocol Link Control and Status registers:
    - por_cxg_ra_cxprtcl_link<X>_ctl
    - por_cxg_ra_cxprtcl_link<X>_status
- Auxiliary Control register (por_cxg_ra_aux_ctl)
- Configuration Control register (por_cxg_ra_cfg_ctl)
- PortID assignment (por_cxg_ra_node_info)

CXHA

- HAID register (por_cxg_ha_id)
- RAID to LDID LUT registers (por_cxg_ha_rnf_exp_raid_to_ldid_reg<X>)
- RAID or HAID to LinkID LUT registers:
    - por_cxg_ha_agentid_to_linkid_reg<X>
    - por_cxg_ha_agentid_to_linkid_val
- CCIX Protocol Link Control and Status registers:
    - por_cxg_ha_cxprtcl_link<X>_ctl
    - por_cxg_ha_cxprtcl_link<X>_status
- Auxiliary Control register (por_cxg_ha_aux_ctl)
- RN SAM

CXLA

- CCIX capabilities (por_cxla_ccix_prop_capabilities). This register is RO.
- CCIX Configured Properties (por_cxla_ccix_prop_configured)
- CXS Interface Properties Registers. These registers are RO:
    - por_cxla_tx_cxs_attr_capabilities
    - por_cxla_rx_cxs_attr_capabilities
- RAID or HAID to LinkID LUT registers:
    - por_cxla_agentid_to_linkid_reg<X>
    - por_cxla_agentid_to_linkid_val

- LinkID to PCIe Bus Number LUT register (por_cxla_linkid_to_pcie_bus_num)
- Auxiliary Control register (por_cxla_aux_ctl)
- Port enable and AgentID to PortID functionality:
    - por_cxla_portfwd_ctl
    - por_cxla_agentid_to_portid_reg<X>
- TLP header field value configuration register (por_cxla_tlp_hdr_fields)

HN-F

- LDID to CHI NodeID registers (por_hnf_rn_cluster<X>_physid_reg<Y>)
- CCIX Port Aggregation Mask register (por_hnf_cml_port_aggr_grp<X>_add_mask)
- CCIX Port Aggregation Control register (por_hnf_cml_port_aggr_grp<X>_reg)

RN-F/RN-I/RN-D

- RN SAM
- CCIX Port Aggregation Mode Enable and Control registers:
    - cml_port_aggr_grp<X>_reg
    - cml_port_aggr_mode_ctrl_reg
- CCIX Port Aggregation Mask register (cml_port_aggr_grp<X>_add_mask)

5.5.2 Bring up a CML system

Use the following sequence to bring up a CML system.

Procedure

1. Discover and bring up the local CMN-650 system.
   For more information, see 5.5.2.1 Discover and bring up a local CMN-650 system on page 851.
2. Discover CCIX devices and CCIX systems.
   For more information, see 5.5.2.2 Discover CCIX devices in CCIX system on page 852.
3. Enumerate and configure CCIX devices.
   For more information, see 5.5.2.3 Enumerate and configure CCIX devices on page 853.

Next steps

For information about the programming requirements that are necessary for communication between CCIX components, see 5.5.3 Program CML system to enable CCIX communication on page 853.

5.5.2.1 Discover and bring up a local CMN-650 system

Use this process to bring up a local CMN-650 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-650 CML system. For the full sequence, see 5.5.2 Bring up a CML system on page 851.

Procedure

1. Complete the CMN-650 discovery mechanism to discover node types, their corresponding locations (node IDs), and their logical IDs.
   4.3.1 Node ID mapping on page 88 and 4.12 Discovery on page 167 define the discovery mechanism. Node types that are relevant to CML-specific programming are RN-Fs, RN-Is, RN-Ds, HN-Fs, and CCIX gateway blocks (CXRA, CXHA, and CXLA).
2. Bring up the local system to allow normal local operations. To bring up all local non-CCIX components (HN-F, HN-I, HN-D, HN-P, RN-I, SN-F, and XP):
   a) Complete CMN-650 boot time programming.
      For more information, see 5.4.1 Boot-time programming sequence on page 837.
   b) Program RN SAM with the local address map.
      For more information, see 5.4.3.1 Program the SAM on page 839.

Next steps

For information about how to discover the CCIX devices in your CML configuration, see 5.5.2.2 Discover CCIX devices in CCIX system on page 852.

5.5.2.2 Discover CCIX devices in CCIX system

CCIX system discovery involves going through the PCIe link activation and device enumeration mechanism. Use this process to discover CCIX devices in your CCIX system.

Before you begin

You must first discover and bring up your local CMN-650 system. For more information, see 5.5.2.1 Discover and bring up a local CMN-650 system on page 851.

About this task

This procedure is the second step in the sequence to bring up a CMN-650 CML system. For the full sequence, see 5.5.2 Bring up a CML system on page 851.

Procedure

Follow the standard PCIe device enumeration steps to detect CCIX capable devices.

For more information, see *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*.

Next steps

If one or more CCIX-capable devices are detected during PCIe device enumeration, then complete the following steps:

- See 5.5.2.3 Enumerate and configure CCIX devices on page 853, which shows how to enumerate CCIX devices.
- Program the PCIe-RC to enable multiple *Virtual Channels* (VCs).

5.5.2.3 Enumerate and configure CCIX devices

Use this procedure to enumerate CCIX devices in your CML system and configure their CCIX properties during bring up.

Before you begin

Before enumerating CCIX devices, you must first complete the following steps:

- Discover and bring up your local CMN-650 system. For more information, see 5.5.2.1 Discover and bring up a local CMN-650 system on page 851.
- Discover CCIX devices in your CCIX system. For more information, see 5.5.2.2 Discover CCIX devices in CCIX system on page 852.

About this task

This procedure is the last step in the sequence to bring up a CMN-650 CML system. For the full sequence, see 5.5.2 Bring up a CML system on page 851.

Procedure

1. Discover all CCIX agents (RA and HA) at each CCIX device.
   These agents must be uniquely identifiable. If any CCIX device contains CMN-650, follow the CMN-650 discovery mechanism to discover node types, their corresponding locations (node IDs), and their logical IDs. The discovery mechanism is described in 4.3.1 Node ID mapping on page 88 and 4.12 Discovery on page 167.
2. Discover the address map requirements of each CCIX device.
3. Read the CCIX capabilities of each CCIX device.
   a) Read the por_cxla_ccix_prop_capabilities register, which is present in each CXLA, to determine the CCIX capabilities of CMN-650.
4. Determine the common properties and capabilities that all CCIX devices support and configure them in each CCIX device.
5. Program the properties that are determined in the preceding step in the por_cxla_ccix_prop_configured register, which is present in each CMN-650 CXLA.

Next steps

To enable CCIX communication, follow the programming procedures that are described in 5.5.3 Program CML system to enable CCIX communication on page 853.

5.5.3 Program CML system to enable CCIX communication

Use this procedure to enable CCIX communication between different CCIX entities. The steps can be completed in any order.

Before you begin

Before enabling CCIX communication, you must first complete the bring up process. For more information, see 5.5.2 Bring up a CML system on page 851.

About this task

The terms *link*, *CCIX link*, and *CCIX protocol link* that are used in subsequent sections refer to CCIX logical link.

The CCIX logical link is defined in the *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*.

Procedure

- Program the auxiliary control and configuration control registers.
  For more information about specific optional functionality, see 5.5.3.1 Options when programming CXG auxiliary control and configuration control registers on page 854.
- Program IDs for local CXRAs and CXHAs. For more information, see 5.5.3.2 Assign IDs for local CXRAs and CXHAs on page 855.
- Program remote CCIX agents:
  - Assign LinkIDs to remote CCIX protocol links. For more information, see 5.5.3.3 Assign LinkIDs to remote CCIX protocol links on page 856.
  - Assign PCIe bus numbers for LinkIDs. For more information, see 5.5.3.5 Assign PCIe bus numbers for LinkIDs on page 857.
  - Assign LDIDs to remote caching agents. For more information, see 5.5.3.6 Assign LDIDs to remote caching agents on page 857.
  - Program RA SAMs. For more information, see 5.5.3.7 Program RA SAM on page 858.
  - Program RN SAM in CXHAs. For more information, see 5.5.3.8 Program RN SAM in CXHA on page 858.
  - Program CCIX protocol link control registers. For more information, see 5.5.3.9 Program CCIX protocol link control registers on page 859.
  - Program CPA functionality in RN SAM, if using CPAGs. For more information, see 5.5.3.11 Program CPA functionality in RN SAM on page 860.
  - Program CPA functionality in HN-F SAM, if using CPAGs. For more information, see 5.5.3.12 Program CPA functionality in HN-F SAM on page 861.
- Enable CCIX port-to-port forwarding, if using this feature.
  For more information, see 5.5.3.13 Enable CCIX port-to-port forwarding on page 862.

5.5.3.1 Options when programming CXG auxiliary control and configuration control registers

CMN-650 has optional CML functionality that can be enabled when you program the CXG auxiliary control and configuration control registers. There are specific constraints that you must follow when enabling this functionality for your CML system.

CMN-650 supports the following functionality in the CXG auxiliary and configuration control registers:

- SMP mode
- CXSA mode

SMP mode

SMP mode is enabled by setting the lnk<X>_smp_mode_en bit in the following registers:

- por_cxg_ra_cxprtcl_link<X>_ctl
- por_cxg_ha_cxprtcl_link<X>_ctl

The SMP mode programming must be the same in CXRA and CXHA for a specific CCIX protocol link. Also, all CXG pairs that can communicate with each other must be configured in the same way.

CXSA mode

CXSA mode is enabled by setting the cxsa_mode_en bit in the por_cxg_ra_cfg_ctl register of the CXRA.

When this mode is enabled, the CXRA inside the CXG is used to communicate with a remote CXSA. In this mode, the CXRA receives requests from local HN-Fs.

5.5.3.2 Assign IDs for local CXRAs and CXHAs

Use this procedure to assign CCIX identifiers for local CXRAs and CXHAs 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 5.5.3 Program CML system to enable CCIX communication on page 853.

Procedure

- Assign *Requesting Agent IDs* (RAIDs) for local CXRAs.
  a) Program all the local RAIDs in the following registers for all CXRAs and set the corresponding valid bit in each CXRA:
     - por_cxg_ra_rnf_ldid_to_exp_raid_reg
     - por_cxg_ra_rni_ldid_to_exp_raid_reg
     - por_cxg_ra_rnd_ldid_to_exp_raid_reg

     This programming sets up the LDID to RAID LUT.
  b) Program the CCIX Source ID (Agent ID) in entry 0 of the por_cxg_ra_rnf_ldid_to_exp_raid_reg0 register and set the corresponding valid bit if CXSA mode is enabled.
- Program *Home Agent IDs* (HAIDs) for all local CXHAs into the por_cxg_ha_id registers that are present in each CXHA.
  This programming is not required if CXSA mode is enabled.
  Because all CXHAs can communicate with all local HNs (HN-F, HN-I, HN-D, and HN-P), they can have the same HAID. However, if uniqueness between HAIDs is required for routing purposes, HAIDs do not have to be the same.

  For compliance with the CCIX specification, a CXHA and CXRA with the same ID must reside behind the same CCIX protocol link. For more information, see *Cache Coherent Interconnect for Accelerators CCIX Base Specification Revision 1.1 Version 1.0*.

5.5.3.3 Assign LinkIDs to remote CCIX protocol links

Use this procedure to assign a unique LinkID to each remote CCIX protocol link with which a CXG can communicate.

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 5.5.3 Program CML system to enable CCIX communication on page 853.

Each CMN-650 CXG contains a CXRA, CXHA, and CXLA node. Each CXG can communicate with up to three remote CCIX protocol links. These links are marked sequentially as links 0, 1, and 2. Remote links are identified using LinkIDs.

Remote CCIX agents (RAs or HAs) are identified using their RAID or HAID. Each remote RA or HA that a CXG can communicate with must be behind only one link.

It is only necessary for LinkIDs to be unique within a CXG. Each CXG has a respective LinkID space. Each remote link has its own CCIX protocol link control and status registers.

Procedure

1. Determine the LinkID of each remote agent, in other words the targets, of the CXG.

2. Program these LinkIDs in the following registers, which are present in the CXRA, CXHA, and CXLA:
    - por_cxg_ra_agentid_to_linkid_reg<X>
    - por_cxg_ha_agentid_to_linkid_reg<X>
    - por_cxla_agentid_to_linkid_reg<X>

    This step sets up the AgentID (RAID or HAID) to LinkID LUT.

    If CXSA mode is enabled for a link, it is not necessary to program the por_cxg_ha_agentid_to_linkid_reg<X> register. The por_cxg_ra_agentid_to_linkid_reg<X> register must be programmed with the CCIX Slave Agent ID for the link.
3. Set the respective valid bits in the following registers:
    - por_cxg_ra_agentid_to_linkid_val
    - por_cxg_ha_agentid_to_linkid_val
    - por_cxla_agentid_to_linkid_val

5.5.3.4 Program AgentID to PortID sets

Use this procedure to program AgentID to PortID sets.

Procedure

1. Determine the target PortID for each target AgentID (CCIX TLP TargetID) that the current port can forward the incoming CCIX TLPs to.
2. Program these PortIDs in the por_cxla_agentid_to_portid_reg<X> register.
    For example, all agents which are local to this chip are assigned the same PortID as the current CCIX port. The LUT indicates that the transaction target is on the current chip, so all the TLPs targeting these agents are consumed locally. They are not forwarded to another port.

5.5.3.5 Assign PCIe bus numbers for LinkIDs

Use this procedure to set up the LinkID to PCIe bus number LUT in the CXLA. This programming is only required if the PCIe header is used to route a CCIX TLP.

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 5.5.3 Program CML system to enable CCIX communication on page 853.

Procedure

- Program the PCIe bus number for each remote link in the por_cxla_linkid_to_pcie_bus_num register, which is present in each CXLA.
- Program the por_cxla_tlp_hdr_fields register if using PCIe headers to route a CCIX TLP.

5.5.3.6 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 CCIX agents. For the full list of requirements, see 5.5.3 Program CML system to enable CCIX communication on page 853.

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.

This programming is not required if CXSA mode is enabled.

Procedure

1. Program unique LDIDs for each remote caching agent in the por_cxg_ha_rnf_exp_raid_to_ldid_reg<X> register that is present in each CXHA. 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_cxg_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 CXHA NodeID at the LDID index of each remote RN-F in por_hnf_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.

5.5.3.7 Program RA SAM

Use this procedure to program the RA SAM, which generates the target ID for CCIX requests.

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 5.5.3 Program CML system to enable CCIX communication on page 853.

Procedure

Program the following properties for each remote HA into the por_cxg_ra_sam_addr_region_reg<X> register that is present in each CXRA, 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

If CXSA mode is enabled, RA SAM must be programmed with the address range and the Agent ID of the remote CCIX Slave Agent.

5.5.3.8 Program RN SAM in CXHA

Use this procedure to program the RN SAM in each CXHA.

About this task

You can program CXHA RN SAM as part of local system bring up.

This programming is not required if CXSA mode is enabled.

Procedure

Program the RN SAM present in each CXHA with the address and memory map of the local HNs.

For more information on RN SAM programming, see 5.4.3 RN SAM and HN-F SAM programming on page 838.

5.5.3.9 Program CCIX protocol link control registers

Use this procedure to set up CCIX protocol links for a CXG and configure the distribution of credits that the CXG uses.

About this task

There is a CCIX protocol link control register for each CCIX protocol link that a given CCIX gateway block (CXRA, CXHA, and CXLA) can communicate with.

Procedure

1. Program the por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_link<X>_ctl registers that are present in each CXRA and CXHA.
   If CXSA mode is enabled, it is not necessary to program the protocol link control registers of the CXHA. In this case, the lnk0_num_snpcrds of the CXRA can be set to 4'hF.
2. Set the lnk<X>_link_en bit for each CCIX protocol link that can be used in the future.
   If this bit is not set, credits are not set aside for this link.
3. 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 5.5.3.10 CCIX protocol link credit distribution on page 859.
   You must ensure that the total percentage of credits that are allocated to all links does not exceed 100.

5.5.3.10 CCIX protocol link credit distribution

When setting up CCIX protocol links, you can specify the CCIX protocol link credit distribution. The distribution can be configured when programming the CCIX protocol link control registers.

The link enable bits (lnk<X>_link_en) of the por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_link<X>_ctl registers determine how many links are active for a CXG. 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 5-507: 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 CCIX protocol link control registers, see 5.5.3.9 Program CCIX protocol link control registers on page 859.

5.5.3.11 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 CCIX agents. For the full list of requirements, see 5.5.3 Program CML system to enable CCIX communication on page 853.

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 5.5.3.12 Program CPA functionality in HN-F SAM on page 861.

There are certain rules and restrictions that apply to how CPA can be enabled in a CMN-650 system. For more information, see 4.8 SAM support for CCIX Port Aggregation on page 126.

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 CXGs 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 CXGs. 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 CXGs for each CPAG by completing the following steps:
   a) Program the number of CXGs 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 CXG by setting the pag0_tgtid field of the same register. If two CXGs 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.

5.5.3.12 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 CCIX agents. For the full list of requirements, see 5.5.3 Program CML system to enable CCIX communication on page 853.

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 5.5.3.11 Program CPA functionality in RN SAM on page 860.

There are certain rules and restrictions that apply to how CPA can be enabled in a CMN-650 system. For more information, see 4.8 SAM support for CCIX Port Aggregation on page 126.

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 por_hnf_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 por_hnf_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, or CHI-D.

   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 CXGs to 1 in the addr_mask field of the por_hnf_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 CXGs. 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 CXGs for each CPAG by completing the following steps:
   a) Program the number of CXGs in each CPAG by setting the num_cxg_pag<n> field in the por_hnf_cml_port_aggr_ctrl_reg register.
   b) Program the CHI node ID of each CXG into the pag_tgtid<n> fields in por_hnf_cml_port_aggr_grp_reg0 and por_hnf_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.

5.5.3.13 Enable CCIX port-to-port forwarding

Use this procedure to enable port-to-port forwarding for your CML configuration. This programming must be completed for each CCIX port in the configuration.

About this task

Each CXG block that is present on CMN-650 is a CCIX port. All CCIX ports are assigned a unique logical ID when Socrates™ builds the CMN-650 configuration. This logical ID is known as the PortID of the CCIX port. You can determine the PortID for a CCIX port by reading the logical ID field (bits [47:32]) of por_cxg_ra_node_info register for the corresponding CXRA.

Enabling port-to-port forwarding for a CML configuration involves PortID assignment and configuration of registers that provide an AgentID to PortID LUT.

The following constraints apply to this process:

- PortIDs are only unique within a chip. In other words, CCIX ports on separate chips might have the same PortID.
- From a given port, all the TLP traffic targeting the same chip must go through the same target port. Failure to follow this constraint violates TLP to TLP ordering guarantees.
- Port credits are evenly distributed between all the communicating ports. Therefore, port_fwd_en must only be set for ports that the current port can receive forwarded traffic from.

Procedure

1. Set the port_fwd_en field in the por_cxla_portfwd_ctl register.
   You must set this field for all CCIX ports that the current port can communicate with. The port_fwd_en field is a 16-bit vector where each bit represents the PortID on CMN-650.
2. Ensure that AgentID to PortID programming is done as described in 5.5.3.4 Program AgentID to PortID sets on page 857.

Next steps

To check the status of the port-to-port link, read the following register bits:

- port_fwd_req field in the por_cxla_portfwd_ctl register
- port_fwd_ack field in the por_cxla_portfwd_status register

The following bit combinations determine the status of the port-to-port link:

Disabled	port_fwd_req = 0 and port_fwd_ack = 0
Activate	port_fwd_req = 1 and port_fwd_ack = 0
Run	port_fwd_req = 1 and port_fwd_ack = 1
Deactivate	port_fwd_req = 0 and port_fwd_ack = 1

5.5.4 Program CMN-650 CML system at runtime

Use this procedure to program a CMN-650 CML system at runtime.

Procedure

1. Bring up CCIX protocol link.
   For more information, see 5.5.5 Establish protocol link up between CXG and remote CCIX link on page 864.
2. Add a CCIX protocol link in system coherency and DVM domains.
   For more information, see 5.5.7 CCIX entry and exit protocol links from coherency domains and DVM domains on page 866.

Note

If CXSA mode is enabled, this programming is not necessary.

3. Program the remote address range and corresponding CXRA node ID for each remote memory region in RN SAM present in CMN-650 RN-F, RN-I, and RN-D.
   RN SAM must not be programmed to target CXRA when enabled for CXSA mode.

Note

If the software can guarantee that there is no traffic to the remote address range until CCIX-related initial programming is complete and CCIX protocol links are up, then this programming should be done when programming RN SAMs with local address map.

5.5.5 Establish protocol link up between CXG and remote CCIX link

Use the following procedure to link up a CXG with a remote CCIX link that the CXG can communicate with.

About this task

A CCIX protocol link can be established between each CXG in CMN-650 and a corresponding remote CCIX link that the CXG can communicate with. The term *link* is used here to refer to a CCIX protocol link. Multiple CCIX links can be set up simultaneously by extending this sequence for each link.

CMN-650 CXGs contain CXRA, CXHA, and CXLA nodes.

Procedure

1. Poll the lnk<X>_link_en bit in both por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_link<X>_ctl 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 por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_link<X>_ctl registers to ensure that they are clear.
   b) Poll the lnk<X>_link_down bits in the por_cxg_ra_cxprtcl_link<X>_status and por_cxg_ha_cxprtcl_link<X>_status registers to ensure that they are set.
   c) Poll the lnk<X>_link_ack bits in the por_cxg_ra_cxprtcl_link<X>_status and por_cxg_ha_cxprtcl_link<X>_status registers to ensure that they are clear.
3. Set the lnk<X>_link_req bits in the por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_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 por_cxg_ra_cxprtcl_link<X>_status and por_cxg_ha_cxprtcl_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 CXRA and CXHA.

   Hardware acknowledgment of link up means that both sides are ready to receive and grant CCIX protocol credits.

5. Set the lnk<X>_link_up bit in the por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_link<X>_ctl registers to instruct both sides to start granting credits.

Results

Link<X> is now up. Both sides can now exchange CCIX protocol credits and protocol messages.

5.5.6 Link down CCIX protocol link between CXG and remote CCIX link

Use this procedure to deactivate a CCIX protocol link between a CMN-650 CXG and a remote CCIX 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 CCIX 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-650:
    1. Poll the lnk<X>_ot_cbkwr bit in the por_cxg_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 5.5.7 CCIX entry and exit protocol links from coherency domains and DVM domains on page 866.
- If CXSA mode is enabled, CCIX links must not be brought down until all HN-Fs communicating with this CXSA are in OFF state.

About this task

You must follow a specific process to bring down a CCIX protocol link between each CMN-650 CXG and the corresponding remote CCIX link that the CXG can communicate with. The term *link* is used here to refer to a CCIX protocol link. Multiple CCIX protocol links can be brought down at the same time by extending this sequence for each link.

CMN-650 CXGs contain CXRA, CXHA, and CXLA nodes.

Procedure

1. Ensure that the link is up and can accept a new link down request by polling the following bits in the por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_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 por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_link<X>_ctl registers to make a link down request.

3. Poll the lnk<X>_link_ack bit in the por_cxg_ra_cxprtcl_link<X>_status and por_cxg_ha_cxprtcl_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 CCIX protocol credits and start returning remote CCIX protocol credits.
4. Poll the lnk<X>_link_down bits in the por_cxg_ra_cxprtcl_link<X>_status and por_cxg_ha_cxprtcl_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 por_cxg_ra_cxprtcl_link<X>_ctl and por_cxg_ha_cxprtcl_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.

5.5.7 CCIX entry and exit protocol links from coherency domains and DVM domains

CMN-650 CXG blocks, which contain CXRA, CXHA, and CXLA nodes, can establish a CCIX protocol link with a remote CCIX link. Each CXG has software-programmable bits to allow the CCIX protocol links to enter and exit the system coherency domains and DVM domains.

The CXHA has 2 bits which facilitate snoop coherency domain entry and exit requests and acknowledgment:

- The lnk<X>_snoopdomain_req bit of the por_cxg_ha_cxprtcl_link<X>_ctl in each CXHA, controls snoop coherency domain requests for the link.
- The lnk<X>_snoopdomain_ack bit of the por_cxg_ha_cxprtcl_link<X>_status register, provides acknowledgment and status of the snoop coherency domain requests for the link.

The CXRA has 2 bits which facilitate DVM domain entry and exit requests and acknowledgment:

- The lnk<X>_dvmdomain_req bit in the por_cxg_ra_cxprtcl_link<X>_ctl register in each CXRA, controls DVM domain requests for the link.
- The lnk<X>_dvmdomain_ack bit in the por_cxg_ra_cxprtcl_link<X>_status register, provides acknowledgment and status of the DVM domain requests for the link.

For more information, see 4.2.9 RN entry to and exit from snoop and DVM domains on page 85.

6 SLC memory system

This chapter describes the optional SLC memory system which is implemented by HN-Fs in the mesh.

6.1 About the SLC memory system

The SLC memory system consists of the HN-F protocol nodes in CMN-650.

There is a configurable number of instances (1-64) of the HN-F. Each HN-F node or slice has the following features:

- 0KB, 128KB, 256KB, 512KB, 1MB, 2MB, 3MB, or 4MB of SLC data RAM and tag RAM.
- Combined *Point-of-Coherency* (PoC) and *Point-of-Serialization* (PoS).
- SF size of 512KB, 1MB, 2MB, 4MB, or 8MB.

Each HN-F in CMN-650 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 and SF are 16-way set-associative. 12-way for 3MB SLC configurations.
- By default the SLC and SF use a pseudo-random victim selection policy. If there is an invalid way, it is not necessary to select a victim.
- Optionally, CMN-650 supports an *enhanced LRU* (eLRU) cache replacement policy that can be enabled by setting a bit in the configuration register. eLRU is Dynamic Biased Replacement Policy. 2 bits per set/way are used to 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.
- SLC and SF arrays:
  - Supports one-cycle, two-cycle, or three-cycle non-pipelined tag array
  - Supports 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

- 32 or 64-entry address and data buffer, which is known as the *PoC Queue* (POCQ), to service:
  - All transactions from the CHI interface
  - SLC evictions to the memory controller
  - SF evictions and associated WriteBacks to the memory controller
- 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 por_hnf_sam_sn_properties register of the HN-F SAM to allow such propagation to each SN-F
- Supports QoS-based protocol flow control:
  - POCQ resources are allocated or rejected for protocol retry according to the QoS class
  - POCQ resources are watermarked for different QoS classes with user-configurable options
  - Starvation prevention for lower-priority QoS classes
  - QoS-based static grantee selection for CHI architecture credit return
- QoS priority-based request selection to the memory controller
- Supports allocation in the 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 address space
- Supports ECC scrubbing for single-bit ECC errors on SF and SLC tag 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. 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 SF only 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

- Supports 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-650 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/way operations. Software specific to CMN-650 would instead be required to flush the SLC, as described in this TRM. Invisible SLC support eliminates the requirement to perform SLC flushes for software context switches from cacheable to non-cacheable.
- Supports up to two memory-region-based SN targets and one, three, five, or six SN-F address hashing
- Supports MPAM

6.2 SLC memory system components and configuration

CMN-650 *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. CMN-650 SLC also 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.

Normally, 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.

6.2.1 HN-F configurable options

The HN-F can be configured in several ways.

The HN-F has the following configurable parameters:

- SLC size of 0KB, 128KB, 256KB, 512KB, 1MB, 2MB, 3MB, or 4MB

- SF size of 512KB, 1MB, 2MB, 4MB, or 8MB
- 16, 32, or 64 POCQ entries
- 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 parameters:

- HN-F CHI interface data-VC (DAT) width of 256 bits

6.2.2 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 but not all. If SF clustered mode is enabled, multicast snoops might be more common.
- Broadcast, to all RN-Fs.

6.2.3 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 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.

6.2.4 HN-F SAM configuration by SN type

CMN-650 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-650 to use the following SN types:

- CHI-C or CHI-D SN-F.
- SBSX.
- CXSA.

To configure SN target types in the HN-F SAM, program the por_hnf_sam_sn_properties register. For the description of this register, see 5.3.15.66 por_hnf_sam_sn_properties on page 784.

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 6-1: por_hnf_sam_sn_properties SN type values

SN type	*<X>*_sn_is_chic bit value	*<X>*_sn_pcmosep_conv_to_pcmo bit value
CHI-C SN-F	`0b1`	`0b1`
CHI-D SN-F	`0b0`	`0b0`
SBSX or CXSA	`0b0`	`0b1`

6.2.5 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-650 SF and SLC.

Various *Address Based Flush* (ABF) configuration registers per HN-F instance support the cache flush engine:

por_hnf_abf_lo_addr ABF lower range address.

por_hnf_abf_hi_addr ABF upper range address.

por_hnf_abf_pr ABF policy register. Triggers flush start, indicates flush operation type.

por_hnf_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-650 SF and SLC. When all cache lines within this range are flushed, a bit in the por_hnf_abf_sr register is set indicating that the flush engine has completed. If enabled, an interrupt (**INTREQPPU**) is then sent.

Note

Interrupt indication and complete bit in the por_hnf_abr_sr registers are set regardless of normal completion or abort condition. To determine if a flush request completed normally or aborted, check the error bits in the por_hnf_abf_sr register.

To complete the flush sequence, the HN-F carries out the following steps:

1. Flush CMN-650 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-650 SLC.
2. Flush the CMN-650 SLC.
3. On completion of the flush:
    a. The HN-F sets the status bit in the por_hnf_abf_sr register when the flush is complete for that HN-F. If there are error conditions, they are also set in the por_hnf_abf_sr register. This register is cleared when 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.
- 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 (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 remain 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 show a summary for SF and SLC caches for all three modes.

Table 6-2: 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 6-3: 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 maintenance of coherency and ordering, 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 por_hnf_abf_pr register is set, until the flush is done. The abf_complete bit of the por_hnf_abf_sr register indicates that the flush is done.
- HN-F must be in one of the three operational modes (FAM, HAM, 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 por_hnf_abf_sr register.
- When ABF completes, check the por_hnf_abf_sr to ensure ABF completed without any errors. If ABF aborted for any reasons, then the por_hnf_abf_sr indicates that the flush was aborted.

6.2.6 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. Any access to those lines is guaranteed to hit in the SLC. The variable memory region is calculated as a factor of the total SLC size and number of ways that are locked. For example, consider an SLC that is built with 16 ways. In this case, 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 6-1: 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 are valid and included in the HN-F way allocation. This definition therefore indicates 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-650 power states

Note

The locked regions do not comprehend Secure as opposed to Non-secure memory regions. Therefore, if aliasing is performed between Secure and Non-secure regions, overlocking can occur.

The following tables specify various combinations of region size and the number of locked ways that software must program. Software can program these values using the hnf_slc_lock_ways register and the hnf_slc_lock_base0 register to hnf_slc_lock_base3 register.

Table 6-4: 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 6-5: 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]

6.2.7 Software-configurable On-Chip Memory

The CMN-650 HN-F supports software configurable *On-Chip Memory* (OCM) which allows for 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-650 power states are not supported in OCM mode.
- All OCM ways must be same across all HN-Fs in a system cache group.
- OCM mode must be enabled before any non-config accesses are sent to HN-F.

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 por_hnf_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 6.2.6 Software-configurable memory region locking on page 873.

Note

Region locking registers do not explicitly control Secure and Non-secure memory regions. Therefore combined Secure and Non-secure memory regions should not exceed the total SLC size that is locked for OCM.

6.2.8 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-650 supports programming of:

- The number of ways that can be locked
- RN devices for which these ways are locked
- 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.

Source-based way locking feature can be enabled by programming the por_hnf_rn_region_lock.rn_region_lock_en bit to 1 in each HN-F instance. The requesting nodes, for which these ways are to be locked must also be explicitly enabled in the por_hnf_rn*region_vec registers. The requesting nodes are individually identified using the logical IDs or cluster IDs. For more information about clustering, see 6.2.12 Configuring clustered mode for SF tracking on page 881. CMN-650 has three RN types: RN-F, RN-I, and RN-D. Each requesting node type has different registers and is uniquely identified in a CMN-650 system using logical IDs. The number of ways that are locked are programmed in the por_hnf_slc_lock_ways.ways field.

The region locking feature has two allocation modes:

- Allocating new cache lines for matching RNs only in the locked ways. By doing so, the matching RNs are restricted to allocate to the locked partition only. This mode can be enabled by setting por_hnf_rn_region_lock.rn_pick_locked_ways_only bit to 1.
- 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.

Note

The HN-F must be in the FAM power state. Source-based cache partitioning is not supported in other CMN-650 power states.

6.2.9 Way-based SLC cache partitioning

Each SLC cache instance can be partitioned into different regions. This partitioning allows 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/RN-D masters can allocate to the corresponding group of SLC ways. By default, all RNs can allocate all 16 ways, as the following tables show.

Table 6-6: Logical RN-F ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-F ID								
				63	62	61	60	----	3	2	1	0
por_hnf_slcway_partition0_rnf_vec	0xC48	[3:0]	{64'{1'b1}}									
por_hnf_slcway_partition1_rnf_vec	0xC50	[7:4]	{64'{1'b1}}									
por_hnf_slcway_partition2_rnf_vec	0xC58	[11:8]	{64'{1'b1}}									
por_hnf_slcway_partition3_rnf_vec	0xC60	[15:12]	{64'{1'b1}}									

Table 6-7: Logical RN-I ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-I ID								
				31	30	29	28	----	3	2	1	0
por_hnf_slcway_partition0_rni_vec	0xC68	[3:0]	{32'{1'b1}}									
por_hnf_slcway_partition1_rni_vec	0xC70	[7:4]	{32'{1'b1}}									
por_hnf_slcway_partition2_rni_vec	0xC78	[11:8]	{32'{1'b1}}									
por_hnf_slcway_partition3_rni_vec	0xC80	[15:12]	{32'{1'b1}}									

Table 6-8: Logical RN-D ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-D ID								
				31	30	29	28	----	3	2	1	0
por_hnf_slcway_partition0_rnd_vec	0xC88	[3:0]	{32'{1'b1}}									
por_hnf_slcway_partition1_rnd_vec	0xC90	[7:4]	{32'{1'b1}}									
por_hnf_slcway_partition2_rnd_vec	0xC98	[11:8]	{32'{1'b1}}									
por_hnf_slcway_partition3_rnd_vec	0xCA0	[15:12]	{32'{1'b1}}									

The registers in these tables are used to mask the ways that are always available for an RN to allocate to. A value of:

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 6.2.12 Configuring clustered mode for SF tracking on page 881.

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 6-1: Reserve ways 0-3 for RN-F {0-3}

1. Write 64'h000000000000000F to por_hnf_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 por_hnf_slcway_partition0_rni_vec. This operation disables all 32 RN-Is from allocating to ways 0-3.
3. Write 32'h0 to por_hnf_slcway_partition0_rnd_vec. This operation disables all 32 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 should 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 32 in the partition mask register. In CML configurations, care must be taken 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.

Note

The HN-F must be in the FAM power state. Cache partitioning is not supported in other CMN-650 power states.

6.2.10 RN-F tracking in the SF

The CMN-650 HN-F SF tracks cache lines that come from specific RN-Fs using an RN-F vector index. Your CMN-650 configuration and the values of some configurable parameters determine the size of this index.

The maximum width of the vector index is 128 bits, and therefore it is limited to 128 entries.

The total width of the SF vector index is calculated using the values of various parameters, using the following calculation:

SF_TOTAL_WIDTH = *SF_MIN_WIDTH* + *SF_RN_ADD_VECTOR_WIDTH*

The configuration calculates *SF_MIN_WIDTH*, using the following calculation:

SF_MIN_WIDTH = ceil((*NUM_LOCAL_RNF* + *NUM_REMOTE_RNF*) / *SF_NUM_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 6.2.11 Non-clustered and clustered mode for SF RN-F tracking on page 879.

6.2.11 Non-clustered and clustered mode for SF RN-F tracking

The CMN-650 SF has two modes for tracking cache lines from RN-Fs: non-clustered and clustered mode. 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-650 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.

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.

Non-clustered mode

To enable non-clustered mode at build-time, set the HN-F *SF_MAX_RNF_PER_CLUSTER* parameter = 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 mapping between the RN-F vector index and the RN-F LDIDs in a configuration with 128 non-clustered RN-Fs.

Table 6-9: 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.

Clustered mode

For the SF to track and identify RN-Fs within cluster groups, the RN-F LDID bits are divided into two separate components: the cluster ID and the device ID. Each cluster ID maps to an entry in the SF vector index. Each device ID identifies an RN within the cluster.

For example, 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.

Note

This example configuration is fully clustered, where all the device IDs in each cluster are assigned. However, you do not have to assign all the device IDs within a cluster group. For more information about the configuration options for clustered mode, see 6.2.12 Configuring clustered mode for SF tracking on page 881.

Table 6-10: 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 6-11: 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
...	...	...	...	...	...	...

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)
63	LDID 504	LDID 505	LDID 506	LDID 507	…	LDID 511

The number of LDID bits that are used for the cluster ID and device ID LDID components depends on two 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 HN-F *SF_MAX_RNF_PER_CLUSTER* parameter

To calculate the width of the device ID component of the LDID, use 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 6-12: 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]

It is important to take the cluster ID and device ID into account when assigning LDIDs to local and remote RN-Fs. Accounting for these values helps you to make sure that RN-Fs are grouped in the cluster as required.

Note

When choosing to use clustered mode, there are some further considerations to make:

- SLC partitioning and SLC way locking using source ID uses 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-650 SF increases the likelihood of SF back invalidations. 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.

6.2.12 Configuring clustered mode for SF tracking

Initially, you configure clustered mode for SF tracking at build time. There are also programmable registers that you can use to configure the runtime behavior of clustered mode.

Build-time configuration options for SF clustered mode

When you build your CMN-650 system, two HN-F build-time configuration parameters affect SF clustered mode: *SF_MAX_RNF_PER_CLUSTER* and *SF_RN_ADD_VECTOR_WIDTH*. The value of *SF_MAX_RNF_PER_CLUSTER* determines whether clustered mode is enabled, and both parameters affect the final width of the SF vector index.

To enable clustered mode for RN-F tracking in the SF, set the *SF_MAX_RNF_PER_CLUSTER* parameter value to a valid value that is >1. This value specifies the maximum number of RN-Fs that are in a cluster group, up to a maximum of eight.

Note

If 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.

Consider a CMN-650 system with 8 RN-Fs and *SF_MAX_RNF_PER_CLUSTER* = 4. The minimum SF vector index width is 2-bits wide in this configuration. Therefore, the SF contains two cluster groups with four RN-Fs per cluster group. The following table shows the LDID mapping of each cluster group to the SF vector index values.

Table 6-13: LDID to SF cluster group mapping for 8 RN-Fs with *SF_MAX_RNF_PER_CLUSTER* = 4

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID 0	LDID 1	LDID 2	LDID 3
1	LDID 4	LDID 5	LDID 6	LDID 7

The system builder can choose to add extra tracking bits in the SF using the *SF_RN_ADD_VECTOR_WIDTH* parameter. For example, if the minimum SF vector index width is 2-bits and *SF_RN_ADD_VECTOR_WIDTH* = 6, then the total SF tracking vector width is extended to 8-bits. Therefore, the SF tracking vector can track up to 32 RN-Fs, in eight cluster groups with four RN-Fs per cluster group. The following table shows the LDID mapping to each cluster group in this configuration.

Table 6-14: LDID to SF cluster group mapping for 2-bit minimum SF width and *SF_RN_ADD_VECTOR_WIDTH* = 6

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
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
3	LDID 12	LDID 13	LDID 14	LDID 15
4	LDID 16	LDID 17	LDID 18	LDID 19

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
5	LDID 20	LDID 21	LDID 22	LDID 23
6	LDID 24	LDID 25	LDID 26	LDID 27
7	LDID 28	LDID 29	LDID 30	LDID 31

You can use the *SF_RN_ADD_VECTOR_WIDTH* parameter to add more cluster groups to your configuration, up to the maximum number of cluster groups that your configuration permits. Software can then reconfigure the clustering scheme to achieve precise tracking of RN-Fs or a hybrid of precise and imprecise tracking.

Runtime configuration options for SF clustered mode

At runtime, SF clustered mode can operate in three ways:

Fully precise tracking

Each cluster only contains one RN-F. This mode is similar to non-clustered mode.

Hybrid tracking

Each cluster can contain any number of RN-Fs up to the value of the *SF_MAX_RNF_PER_CLUSTER* parameter.

Fully clustered tracking

Each cluster contains the maximum number of RN-Fs, which the *SF_MAX_RNF_PER_CLUSTER* parameter specifies.

Note

To use fully precise or hybrid SF tracking, you must use the *SF_RN_ADD_VECTOR_WIDTH* parameter to add extra tracking bits to the SF. For more information, see Build-time configuration options for SF clustered mode on page 882.

Default RN-F LDID mapping to SF clusters at reset

Out of reset, the MXPs on a chip assign sequential LDIDs to the RN-Fs in each chip. Consider a CMN-650 system with eight RN-Fs, *SF_MAX_RNF_PER_CLUSTER* = 4, and *SF_RN_ADD_VECTOR_WIDTH* = 6. This configuration has eight cluster groups.

If all eight RN-Fs are on the same chip, the MXP assigns them LDIDs 0-7. The following table shows the default mapping of the LDIDs to SF clusters for this configuration. Only clusters 0 and 1 are used and clusters 2-7 are unused.

Table 6-15: Default LDID mapping to SF clusters

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID 0	LDID 1	LDID 2	LDID 3
1	LDID 4	LDID 5	LDID 6	LDID 7
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
5	-	-	-	-
6	-	-	-	-
7	-	-	-	-

In all modes, you must program the attributes of each valid LDID in the relevant por_hnf_rn_cluster_[0-127]_physid_reg[0-3] registers. Each register contains the properties for two RNs. The programming of these registers must be consistent in each chip. The HN-F uses this information to issue snoop requests. These registers contain attributes for each RN in the system such as:

- NodeID
- CHI source type
- Whether the RN is local or remote
- *CCIX Port Aggregation* (CPA) information
- Whether the contents of the register are valid

In the preceding table, each LDID entry maps to one of the por_hnf_rn_cluster_[0-127]_physid_reg[0-3] registers.

Note

In this example, only the por_hnf_rn_cluster_[0-1]_physid_reg[0-1] registers must be programmed, because there are only eight RN-Fs, divided into two cluster groups.

The por_mxp_device_port_connect_ldid_info_p_[0-1] register contains the default LDID assignment of RN-Fs that are either directly attached to each MXP device port or attached behind a CAL.

In the default tracking configuration, the SF tracks RN-F shared cache lines at the cluster level. Therefore tracking is imprecise and the HN-F snoops all the RN-Fs in that cluster when required.

Fully precise tracking

After reset, you can program each cluster to have one RN-F so that the SF can track each RN-F precisely. To support this reconfiguration, each MXP has por_mxp_p_[0-1]_ldid_override registers. These registers override the default LDID assignment of RN-Fs that are either attached directly to each MXP device port or attached behind a CAL.

Note

The reset value of the por_mxp_p_[0-1]_ldid_override registers is the default LDID assignment. These registers are only present if clustering is enabled and one or more RN-Fs are connected to this device port.

For example, to set a system with eight RN-Fs to use fully precise tracking, program the LDID override values to match the following table.

Table 6-16: Fully precise LDID mapping to SF clusters

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID 0	-	-	-
1	LDID 4	-	-	-
2	LDID 8	-	-	-
3	LDID 12	-	-	-
4	LDID 16	-	-	-
5	LDID 20	-	-	-
6	LDID 24	-	-	-
7	LDID 28	-	-	-

Hybrid tracking

You can also program the SF clustering to use a hybrid of precise and imprecise tracking. If the configuration uses hybrid tracking, the number of RN-Fs in each cluster can be different from the other clusters. To set up a hybrid configuration, program the por_mxp_p_[0-1]_ldid_override registers to match the required LDID mapping. For example, the following table shows an example mapping for a configuration with eight RN-Fs and eight cluster groups.

Table 6-17: Hybrid LDID mapping to SF clusters

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID 0	LDID 1	-	-
1	LDID 4	-	LDID 6	-
2	LDID 8	-	-	-
3	LDID 12	-	-	-
4	LDID 16	-	-	-
5	LDID 20	-	-	-
6	-	-	-	-
7	-	-	-	-

In this hybrid configuration, the SF tracks shared cache lines in clusters 0 and 1 imprecisely. It tracks clusters 2-5 precisely. Clusters 6 and 7 are unused.

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 in use.

Local and remote RN-Fs can be combined into the same cluster.

Fully clustered tracking

If the configuration uses fully clustered tracking, every cluster contains as many valid RN-Fs as the *SF_MAX_RNF_PER_CLUSTER* parameter specifies. For example, in a system with 32 RN-Fs and *SF_MAX_RNF_PER_CLUSTER* = 4, the SF width is 8-bits wide. Therefore the LDID mapping to SF clusters is fully populated.

Table 6-18: Fully clustered LDID to SF cluster mapping

RN-F vector index value (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
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
3	LDID 12	LDID 13	LDID 14	LDID 15
4	LDID 16	LDID 17	LDID 18	LDID 19
5	LDID 20	LDID 21	LDID 22	LDID 23
6	LDID 24	LDID 25	LDID 26	LDID 27
7	LDID 28	LDID 29	LDID 30	LDID 31

In this fully clustered configuration, the SF tracks shared cache lines at the cluster level. Therefore tracking is imprecise and the HN-F snoops all the RN-Fs in that cluster when required.

Other programming requirements for SF clustered mode

When you are programming a CML system to enable CCIX communication, you must assign *Requesting Agent IDs* (RAIDs) to local RNs, using the following registers:

- por_cxg_ra_rnf_ldid_to_exp_raid_reg [0-127]
- por_cxg_ra_rni_ldid_to_exp_raid_reg [0-9]
- por_cxg_ra_rnd_ldid_to_exp_raid_reg [0-9]

This step sets up the LDID to RAID *Lookup Table* (LUT) in the CXRA. For hybrid and fully precise tracking, you override the LDIDs of RN-Fs in the MXP. Therefore, you must program the overridden value for the LDIDs in the por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg_[0-127] registers.

In the CXHA, you must also program unique LDIDs for each remote caching agent in the por_cxg_ha_rnf_ep_raid_to_ldid_reg_[0-255] registers. 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 the LDIDs are overridden.

Related information

- 2.5 Device-level configuration parameters on page 25
- 5.3.15.97 por_hnf_rn_cluster0-63_physid_reg0 on page 823
- 5.3.15.98 por_hnf_rn_cluster64-127_physid_reg0 on page 826
- 5.3.15.99 por_hnf_rn_cluster0-127_physid_reg1 on page 829
- 5.3.15.100 por_hnf_rn_cluster0-127_physid_reg2 on page 831
- 5.3.15.101 por_hnf_rn_cluster0-127_physid_reg3 on page 834
- 5.3.14.2 por_mxp_device_port_connect_info_p0-1 on page 650
- 5.3.14.12 por_mxp_p0-1_ldid_override on page 662

- 5.3.2.15 por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127 on page 281
- 5.5.3 Program CML system to enable CCIX communication on page 853
- 5.5.3.2 Assign IDs for local CXRAs and CXHAs on page 855

6.3 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 tag RAM and double-bit ECC error detection in SLC data RAM

Uncorrectable errors

For example, double-bit ECC errors in the SLC 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 4.17 Reliability, Availability, and Serviceability on page 184.

For information regarding the error source, see the ERRSRC field of 5.3.15.25 por_hnf_errmisc on page 740.

6.3.1 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 is the source of the data is defined as a system cache hit. If error injection and reporting are enabled, any system cache hit drives the slave 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. SLC misses do not drive any errors or error interrupts. If enabled, this mechanism causes only an error to be logged and optionally an interrupt to be generated. 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 5.3.15.26 por_hnf_err_inj on page 742.

6.3.2 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 5.3.15.27 por_hnf_byte_par_err_inj on page 743. 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 DataCheck field of the DAT flit that is returned to an RN is altered.

6.4 Transaction handling in SLC memory system

The CMN-650 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.

6.4.1 Cache maintenance operations

CMN-650 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 snoopable, 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.

6.4.2 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.
- 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*.

Note

Each HN-F in CMN-650 can support tracking of up to 512 logical processors for exclusive operations. In configurations with up to 64 RN-Fs, HN-F supports 64 exclusive monitors and 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. The maximum number of exclusive monitors is 512. The system programmer must ensure that there are no more logical processors capable of concurrently sending exclusive operations than the number of exclusive monitors.

6.4.3 DataSource handling

CMN-650 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 PrefetchTgt (memory controller prefetch) transaction.
- To profile and debug software to evaluate and optimize data sharing patterns.

Table 6-19: DataSource encodings

Source of data	Message	Encoding
HN-I	Default (non-memory source)	0b0000

Source of data	Message	Encoding
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	PrefetchTgt was useful.	`0b0110`
SN-F or SBSX	PrefetchTgt was not useful.	`0b0111`

CMN-650 drives the DataSource value only when the source of the data is either HN-F or HN-I. For other data sources, CMN-650 acts as a conduit.

The encoding that CMN-650 uses 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.

6.4.4 CMO and PCMO propagation from HN-F to SN-F or SBSX

CMN-650 supports propagation of CMO and PCMO requests for a given cache line to the memory controller. The completion point of these requests is programmable.

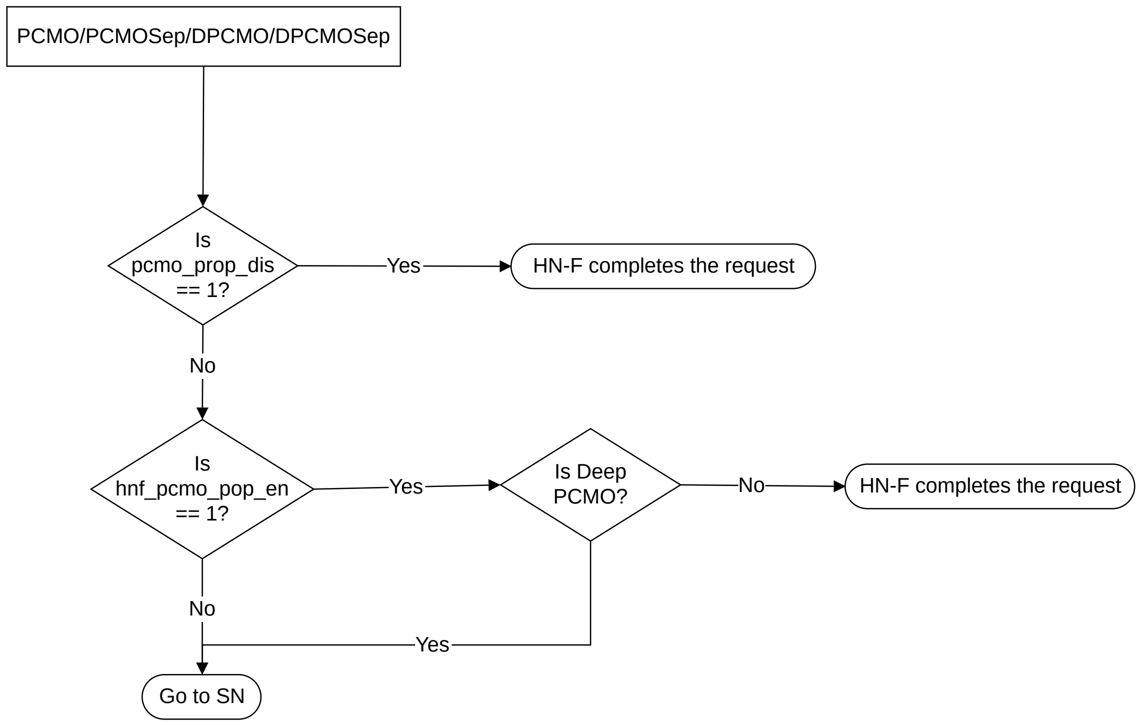
This feature ensures that the cache line has been written to the memory controller and any copies in the CMN-650 system have been removed. Conditional CMO and PCMO propagation to the memory controller also supports external DRAM caches. This feature can be enabled or disabled in the por_hnf_sam_sn_properties register bits for each HN-F corresponding to each SN-F. For SBSX with an AXI4 slave 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:

- por_hnf_sam_sn_properties.pcmo_prop_dis
- por_hnf_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 6-2: PCMO and PCMOSep completion point flow diagram



If both of the following circumstances are true, you must set the por_hnf_sam_sn_properties.X_sn_pcmosep_conv_to_pcmo bit to 1:

- An HN-F is programmed to propagate PCMO requests to an SN-F
- The SN-F does not support PCMOSep requests

This setting allows 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.

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 por_hnf_sam_sn_properties register.

Table 6-20: CMO propagation programming in HN-F SAM

HN-F SAM attribute (cmo_prop_en)	CMO completion point
0	HN-F
1	SN

See the following register descriptions for more information:

- 5.3.15.66 por_hnf_sam_sn_properties on page 784

- 5.3.15.6 por_hnf_cfg_ctl on page 714

6.4.5 Memory System Performance Resource Partitioning and Monitoring

CMN-650 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*

6.4.5.1 MPAM propagation

When MPAM is enabled, CMN-650 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
- CXRA and CXHA
- SBSX

Note

The *`AXMPAM_EN`* parameter controls if MPAM bits are stored and propagated on RN-I, HN-I, and SBSX bridges.

Note

If MPAM support is required in CMN-650, set the *`CHI_MPAM_ENABLE`* parameter.

HN-F behavior

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.

MXP behavior

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. For more information, see 5.3.14.11 por_mxp_p0-1_mpam_override on page 661.

RN-I behavior

When MPAM is enabled, the RN-I has the following extra components:

- Two extra signals on the AXI or ACE-Lite slave 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.

CXRA behavior

The CXRA node type has the following MPAM modes:

SMP mode

The CXRA passes the MPAM field on the USER field of the request. When snooped, the CXRA receives the MPAM field and passes it through the CHI SNP MPAM field.

Non-SMP mode

The CXRA drops the MPAM field that is received on the CHI request. The CXRA also does not receive MPAM field on CGL snoop in this mode.

CXSA mode

The CXRA 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.

CXHA behavior

The CXHA node type has the following MPAM modes:

SMP mode

The CXHA receives MPAM fields through the USER field of the request and passes them through the CHI MPAM field. On incoming CHI transactions, the CXHA passes CHI MPAM values through on the USER field.

Non-SMP mode

The CXHA drops the MPAM values that it receives on CHI snoop. The CXHA does not receive the MPAM attributes on CGL request.

HN-I and SBSX behavior

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-650, MPAM support is applicable to the SLC.

6.4.5.2 MPAM configuration

CMN-650 provides several configuration parameters to configure MPAM features that are used by the interconnect. The MPAM feature ID register, por_hnf_mpam_idr, provides information on what MPAM features 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 2.4 Global configuration parameters on page 23 and 2.5 Device-level configuration parameters on page 25.

For more information about supported MPAM features, see 5.3.12.3 por_hnf_mpam_idr on page 594.

For more information about the software-programmable MPAM override mechanism, see 6.4.5.6 Software-programmable MPAM override on page 896.

6.4.5.3 Cache portion and capacity partitioning

MPAM support in the interconnect is for the CMN-650 SLC. If MPAM is enabled, cache replacement is forced to use an *enhanced Least Recently Used* (eLRU) cache replacement policy.

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:

- Number of portions is the same as the number of ways in SLC.
- The hnf_mpam_ccap_idr_cmax_wd field of the por_hnf_mpam_ccap_idr register is set to 6. This setting provides granularity of 1.56% (1 / 2^6) 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-650 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-650 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 por_hnf_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.

6.4.5.4 Cache capacity monitoring

This section 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 por_hnf_X_msmon_cfg_csu_flt and por_hnf_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 por_hnf_X_msmon_cfg_csu_ctl register supports external capture events 6 and 7:

- External capture event 6 is triggered using **PMUSNAPSHOT** interface or ss_req config bit.
- External capture event 7 is triggered as described in the *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A*.

Note

Multiple capture events cannot be triggered within 32 cycles of each other.

6.4.5.5 MPAM error logging and reporting

CMN-650 implements programmable registers that can enable, disable, and modify MPAM error logging and reporting behavior.

The MPAM error status register, por_hnf_X_mpam_esr, and the MPAM error control register, por_hnf_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 por_hnf_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).

Except where noted in these exceptions, CMN-650 supports MPAM error codes 0-5. For more information about MPAM errors, see *Section 12.2, 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:

- 5.3.12.15 por_hnf_ns_mpam_ecr on page 607
- 5.3.12.16 por_hnf_ns_mpam_esr on page 608
- 5.3.8.5 por_hnf_s_mpam_ecr on page 483
- 5.3.8.6 por_hnf_s_mpam_esr on page 484

6.4.5.6 Software-programmable MPAM override

CMN-650 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
- CXHA
- MXP

Override mechanism in RN-I, RN-D, and CXHA

RN-Is, RN-Ds, and CXHAs 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.

Override mechanism in MXP

The MXP software MPAM override mechanism is useful in the following situations:

- RN-F is a device that is compliant with CHI-D, 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 and por_mxp_p1_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 6-21: por_mxp_p{0,1}_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 requirements:

- 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 combination of MPAM.NS = 1 and REQ.NS = 0 is not valid. In this case, the NS field of the REQ flit drives the MPAM NS field to 0.

The override behavior in the MXP depends on the CHI version that the RN-F supports and whether the enable bit is set.

When *CHI_MPAM_ENABLE* = 1, the RN-F CHI version is CHI-B or CHI-C, and the override enable bit is not set, then the override mechanism has the following behavior:

- All MPAM fields in the REQ flit, except the MPAM NS field, are driven to the default value, which is `0x00`.
- The MPAM NS value is driven to the value of the NS field in the REQ flit.

If all the following conditions are true, then all fields, including the MPAM NS field, are overridden:

- *CHI_MPAM_ENABLE* = 1
- The RN-F CHI version is CHI-B or CHI-C
- The override enable bit is set

The only exception is if the MPAM NS field = 1 and the REQ NS field = 0. In this case, the REQ NS field drives the MPAM NS field to 0.

If all the following conditions are true, then the request uses the original MPAM values of the REQ flit:

- *CHI_MPAM_ENABLE* = 1
- The RN-F CHI version is CHI-D
- The override enable bit is not set

If all the following conditions are true, then all MPAM fields are overridden with the contents of the register:

- *CHI_MPAM_ENABLE* = 1
- The RN-F CHI version is CHI-D
- The override enable bit is set

The only exception is if the MPAM NS field = 1 and the REQ NS field = 0. In this case, the REQ NS field drives the MPAM NS field to 0.

6.5 QoS features

The HN-F protocol queue (POCQ) is a key shared system resource that communicates with the memory controller for external memory access.

The HN-F provides QoS capabilities in support of the following traffic classes:

- Real-time or pseudo-real-time traffic that requires a maximum bounded latency at potentially fixed bandwidth.
- Latency-sensitive traffic, traditionally from a processor device.

CMN-650 uses QoS values to designate these traffic classes. Every request to the HN-F has a 4-bit QoS value that is associated with it, with a higher number indicating a higher priority. The four QoS classes are:

- Highest priority (known as HighHigh).
- High priority.
- Medium priority.
- Low priority.

6.5.1 QoS decoding

QoS decoding takes place inside the HN-F.

The QoS decoding is as follows:

- The CHI interface supports a 4-bit QoS value.
- The 4-bit QoS has 16 possible values. For the QoS ranges and class values in HN-F, refer to Table 4-67: QoS classes in HN-F on page 217.
- QoS mapping is fixed, and is shown in the qos_band register.

The POCQ is logically partitioned to service different QoS class traffic. The HN-F also uses the priorities in the table to arbitrate for the following:

- Memory controller request selection in the POCQ control block.
- Data return selection logic, that is, a CompData to a requester.
- Protocol credits that are sent to an RN-F or RN-I following a protocol-layer retry.

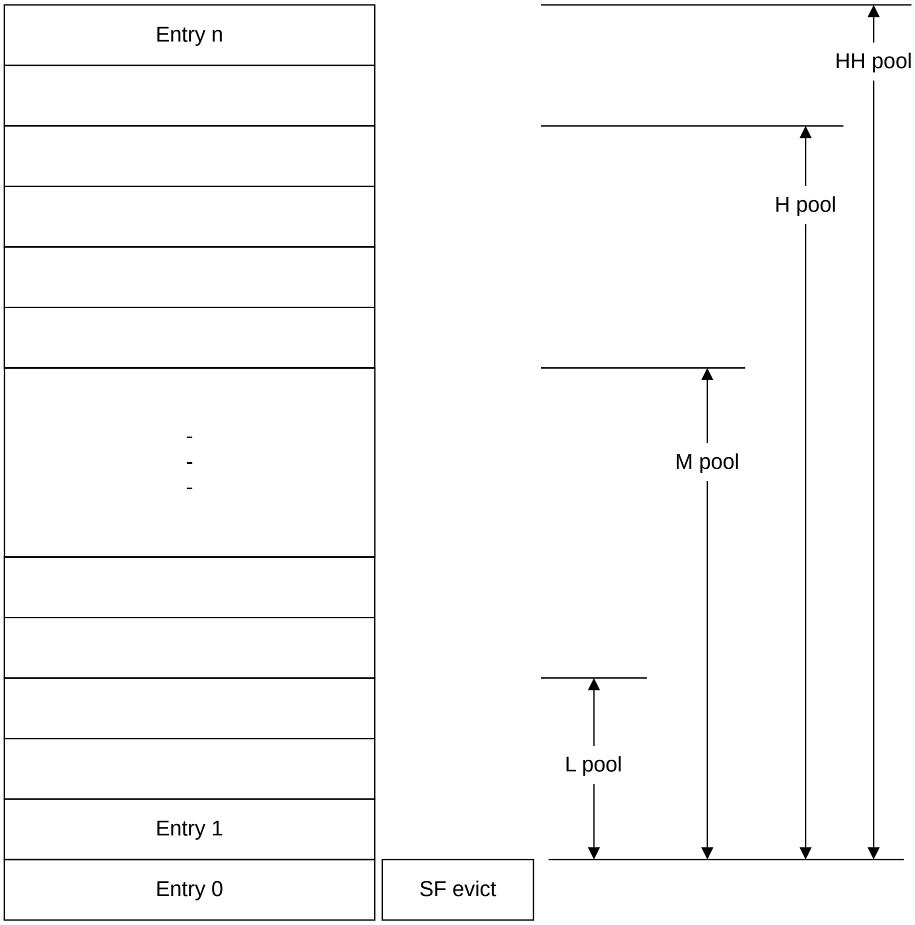
6.5.2 QoS class and POCQ resource availability

The POCQ buffers are shared resources for all QoS classes.

The higher the QoS class, the higher the occupancy availability. For example, the *HighHigh* (HH) QoS class can use all the POCQ entries except for the dedicated SF pool.

The following figure shows the availability of POCQ resources for various QoS levels, using a particular QoS pool that is shared between multiple QoS classes.

Figure 6-3: POCQ availability and QoS classes



The QoS pools are:

hh_pool	Available for HH class.
h_pool	Available for H class and HH class.
m_pool	Available for M class, H class, and HH class.
l_pool	Available for all classes.
seq	SF evictions only.

Warning

Register fields must be programmed so that highhigh_qos_max_cnt > high_qos_max_cnt > med_qos_max_cnt > low_qos_max_cnt ≥ 2. The maximum value that is allowed for highhigh_qos_max_cnt = (*NUM_ENTRIES_POCQ* - seq_qos_max_cnt).

This scheme enables a higher-priority QoS class to have more POCQ resources for transaction processing, and prevents a lower-priority QoS from using all the POCQ. The level of POCQ availability decreases for the lower QoS classes.

QoS pool distribution of the POCQ is software-configurable using the qos_reservation register.

7 Debug trace and PMU

This chapter describes the *Debug Trace* (DT) and *Performance Monitoring Unit* (PMU) features that CMN-650 implements.

7.1 Debug trace system overview

CMN-650 provides at-speed self-hosted *Debug Trace* (DT) capabilities.

The CMN-650 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-650 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 master DTC, have an ATB interface and the following signals:

- **DBGWATCHTRIGREQ**
- **DBGWATCHTRIGACK**
- **INTREQPMU**

The following signals are only present in the master DTC:

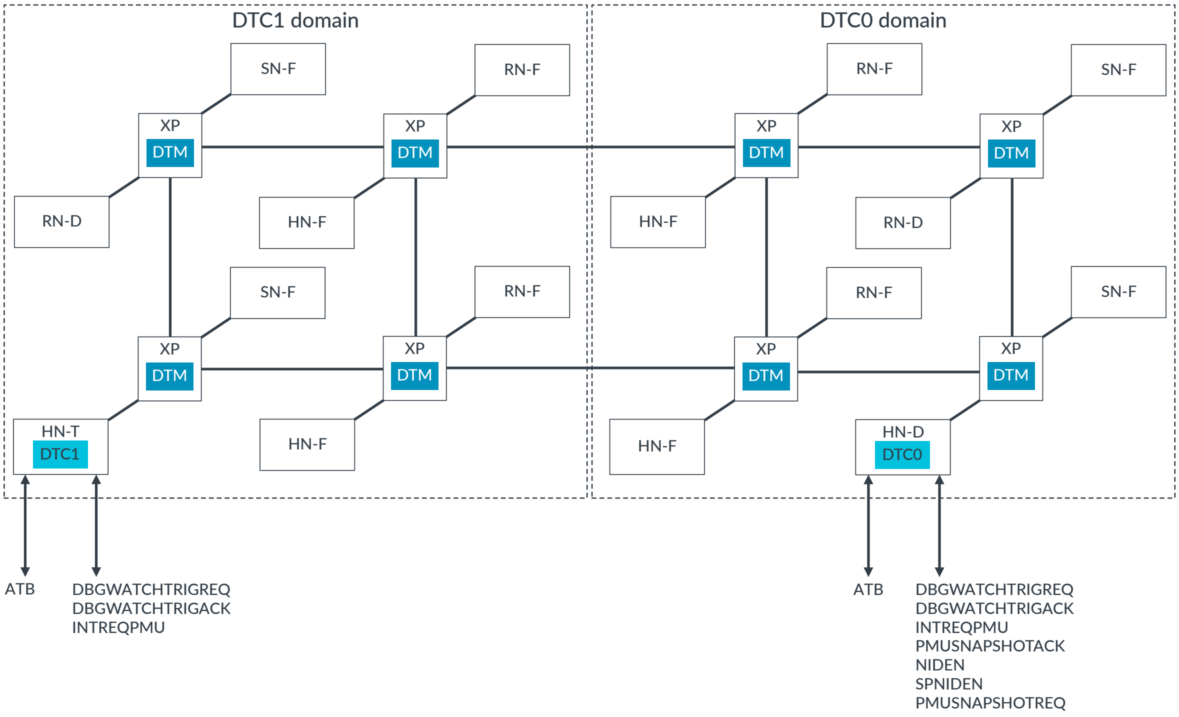
- **NIDEN**
- **SPNIDEN**
- **PMUSNAPSHOTREQ**
- **PMUSNAPSHOTACK**

Note

NIDEN and **SPNIDEN** are propagated from the master 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-650 components transit into their debug and trace states correctly.

The following figure shows an example DT system with two DTC domains.

Figure 7-1: Example DT system with two DTC domains



Note

We recommend one DTC domain per 16 XPs. A combined maximum of 63 XP and CXRH DTMs and PMUs are permitted in a single DTC domain. For a system with more than 63 XPs, at least one HN-T is required.

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 master DTC or DTC0. You assign DTMs to DTCs by configuring XP parameters in Socrates IP Tooling platform.

Each DTC domain must be built using contiguous XPs.

The DT system implements the following functions:

- Monitoring of 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

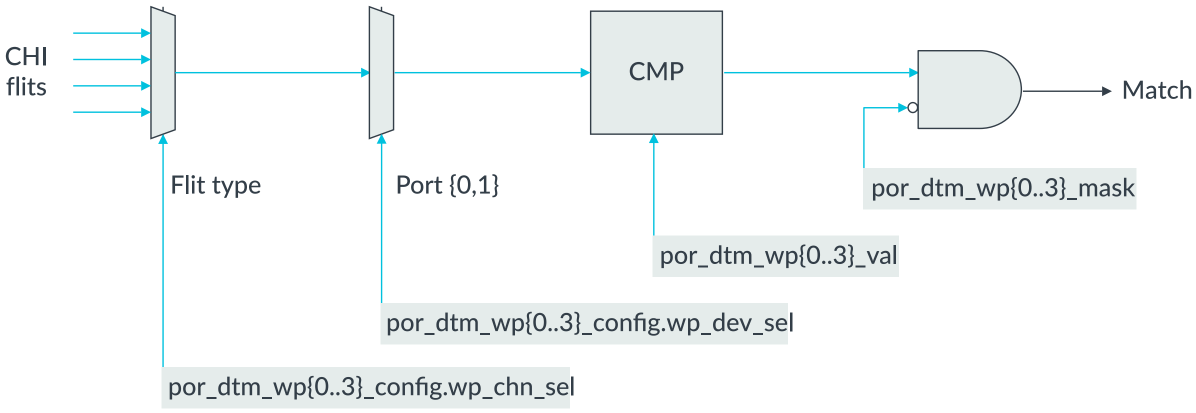
- Trace tag generation
- Debug trigger signaling and trace packet streaming over the ATB at each DTC
- Internal event-based cross trigger generation and broadcast to all DTMs
- Globally synchronized cycle counters

7.1.1 DTM watchpoint

A DTM has four 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 7-2: DTM WP comparator



A WP can be configured to monitor flits from one of two XP device ports and one of four CHI channels:

- REQ
- RSP
- SNP
- DAT

In addition, you can configure the WP to perform 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

You can combine two WPs within a group for complex matching. For example, you can combine WP0 and WP1, just as you can combine WP2 and WP3.

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 5.4.7.1 Program DTM watchpoint on page 846 for the DTM watchpoint programming sequence.

7.1.1.1 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.

The por_dtm_wpN_config registers also define some of the WP behavior.

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. For example, consider if both Opcode and Addr fields of flits that are uploaded on the REQ channel are to be matched. In this situation, WP0 and WP1 must be combined, with Opcode match specified in WP0 and Addr match specified in WP1 or the opposite way. Similarly, consider if both Opcode and Addr fields of flits downloaded on the REQ channel must match. In this case, WP2 and WP3 must be combined in a similar way.

The following table shows REQ channel width and bit ranges for the primary match group.

Table 7-1: REQ channel: primary match group

Field	Width	Bit range
SRCID/TGTID	11	10:0
STASHNID/RETURNNID/Reserved[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
Reserved	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 7-2: REQ channel: secondary match group

Field	Width	Bit range
QOS	4	3:0
ADDR	52	55:4
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 7-3: REQ channel: tertiary match group

Field	Width	Bit range
SRCID/TGTID	11	10:0
OPCODE	7	17:11
MPAM	11	28:18
Reserved	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 7-4: 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
Reserved	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 7-5: 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]
FWDNID	11	[29:19]
OPCODE	5	[34:30]
NS	1	[35:35]
DONOTDATAPULL /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 7-6: 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 7-7: 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 7-8: DAT channel: secondary match group

Field	Width	Bit range
SRC/TGTID	11	10:0
OPCODE	4	14:11
RESPERR	2	16:15
RESP	3	19:17
Reserved	2	21:20
Reserved	8	29:22
Reserved	2	31:30
DBID	12	43:32
TRACETAG	1	44:44
CHUNKV	2	46:45
DEVEVENT	2	48:47
RSVDC	8	56:49

7.1.2 DTM FIFO buffer

Traces captured by the DTM are stored in a four-entry DTM FIFO buffer. Each entry is 160-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.

7.1.2.1 Trace data format

CMN-650 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 7-9: Trace data formats

Packet type	Trace data format		Size	Max traces per FIFO entry
000	TXNID[9:0]		10 bits	16
001	{OPCODE[5:0],TXNID[9:0]}		16 bits	10
010	{2'b00{TGTID[10:0], SRCID[10:0], OPCODE[5:0],TXNID[9:0]}		40 bits	4
011	Reserved		-	-
100	Control flit (see the following tables for field descriptions)	REQ	160 bits	1
		RSP	68 bits	
		SNP	115 bits	
		DAT	94 bits	
101	DATA[127:0]		-	-
110	DATA[255:128]		-	-
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 7-10: 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	10	35:26	

Field	Width	Bit range (MPAM enabled)	Bit range (MPAM disabled)
STASHTGTID / RETURNNID / Reserved[6:0]	11	46:36	
STASHTGTVALID /ENDIAN	1	47:47	
RETURNTID[9:0] / {4'b0, STASHLPVALID, STASHLP[4:0]}	10	57:48	
OPCODE	6	63:58	
SIZE	3	66:64	
NS	1	67:67	
LIKELYSHARED	1	68:68	
ALLOWRETRY	1	69:69	
ORDER	2	71:70	
PCRDTYPE	4	75:72	
MEMATTR	4	79:76	
SNPATTR	1	80:80	
LPID	5	85:81	
EXCL/SNOOPME	1	86:86	
EXPCOMPACK	1	87:87	
TRACETAG	1	88:88	
MPAM	11	99:89	-
ADDR	52	151:100	140:89
RSVDC	8	159:152	148:141
Total	160 (MPAM enabled) / 149 (MPAM disabled)	-	

The following table contains RSP control flit information.

Table 7-11: RSP control flit

Field	Width	Bit range
QOS	4	3:0
TGTID	11	14:4
SRCID	11	25:15
TXNID	10	35:26
OPCODE	4	39:36
FWDSTATE / DATAPULL	3	42:40
RESPERR	2	44:43
RESP	3	47:45
CBUSY	3	50:48
DBID	10	60:51
PCRDTYPE	4	64:61
TRACETAG	1	65:65
DEVEVENT	2	67:66
Reserved	1	68:68

Field	Width	Bit range
Total	69	-

The following table contains SNP control flit information, when MPAM is either enabled or disabled.

Table 7-12: SNP control flit

Field	Width	Bit range (MPAM enabled)	Bit range (MPAM disabled)
QOS	4	3:0	
SRCID	11	14:4	
TXNID	10	24:15	
FWDNID	11	35:25	
FWDTXNID[9:0] / {4'b0, STASHLPVALID, STASHLP[4:0]} / {2'b00VMIDEXT[7:0]}	10	45:36	
OPCODE	5	50:46	
NS	1	51:51	
DONOTGOTOSD / DONOTDATAPULL	1	52:52	
RETTOSRC	1	53:53	
TRACETAG	1	54:54	
MPAM	11	65:55	-
ADDR	49	114:66	103:55
Total	115 (MPAM enabled) / 104 (MPAM disabled)	-	

See also 8.11 DEVEVENT on page 949 for more information.

The following table contains DAT control flit information.

Table 7-13: DAT control flit

Field	Width	Bit range
QOS	4	3:0
TGTID	11	14:4
SRCID	11	25:15
TXNID	10	35:26
HOMENID	11	46:36
OPCODE	4	50:47
RESPERR	2	52:51
RESP	3	55:53
FWDSTATE / DATAPULL	4	59:56
CBUSY	3	62:60
DBID	10	72:63
CCID	2	74:73
DATAID	2	76:75

Field	Width	Bit range
TRACETAG	1	77:77
POISON	4	81:78
CHUNKV	2	83:82
DEVEVENT	2	85:84
RSVDC	8	93:86
Total	94	-

See 8.11 DEVEVENT on page 949 for more information.

Note

CHUNKV[1:0] denotes whether the upper or lower 128 bits of data are valid.

7.1.3 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 TGTID are accumulated in the FIFO. The oldest transactions are on 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, 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 of 1 clears the status bit and enables the corresponding FIFO entry to capture subsequent trace data.

7.1.4 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.

7.1.5 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 flit trace data from the DTMs into the DTC trace FIFO, 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:

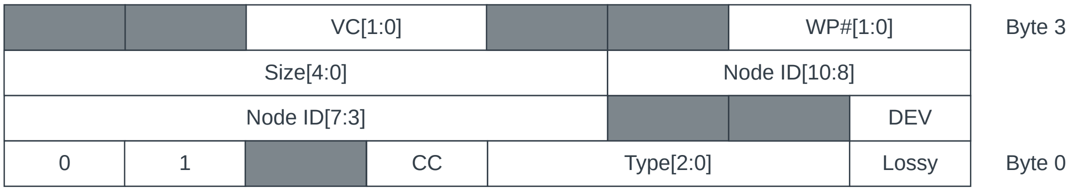
- 7.1.5.1 Trace data packet format on page 913
- 7.1.5.2 Alignment sync packet format on page 914
- 7.1.5.3 Time stamp packet format on page 915
- 7.1.5.4 Cycle counting packet format on page 915
- 7.1.5.5 Trace stream example on page 916

7.1.5.1 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 7-3: Trace data packet header



<table>
<tr><td></td><td></td><td colspan="2">VC[1:0]</td><td></td><td></td><td colspan="2">WP#[1:0]</td><td>Byte 3</td></tr>
<tr><td colspan="5">Size[4:0]</td><td colspan="3">Node ID[10:8]</td><td></td></tr>
<tr><td colspan="5">Node ID[7:3]</td><td></td><td></td><td>DEV</td><td></td></tr>
<tr><td>0</td><td>1</td><td></td><td>CC</td><td colspan="3">Type[2:0]</td><td>Lossy</td><td>Byte 0</td></tr>
</table>

The packet header contains the following fields:

VC CHI channel

00 REQ

01 RSP

10 SNP

11 DAT

DEV Device port number (0 or 1)

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, shifted right by 3 bits reflecting the (X,Y) coordinates of the XP where the trace was captured

CC Cycle counter. When set, this field indicates that the packet after the payload includes a 2B cycle count.

The following key points must be observed:

- For packet type 100, 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). With the cycle counter, another 2B are included at the end of the trace data.
- Whenever 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.

7.1.5.2 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.

7.1.5.3 Time stamp packet format

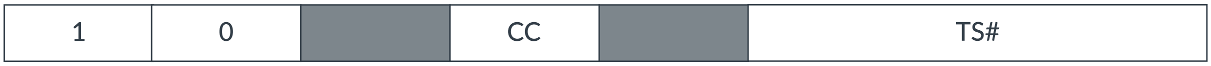
The time stamp packet carries the SoC timer information. The time stamp is used to align the sequence of trace events across the SoC.

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
- When flush is complete
- Periodically based on the setting of the timestamp_period field of the por_dt_trace_control register and only when trace packets have been sent after the last time stamp packet

The following figure shows the time stamp packet format.

Figure 7-4: Time stamp packet



The time stamp packet contains the following fields:

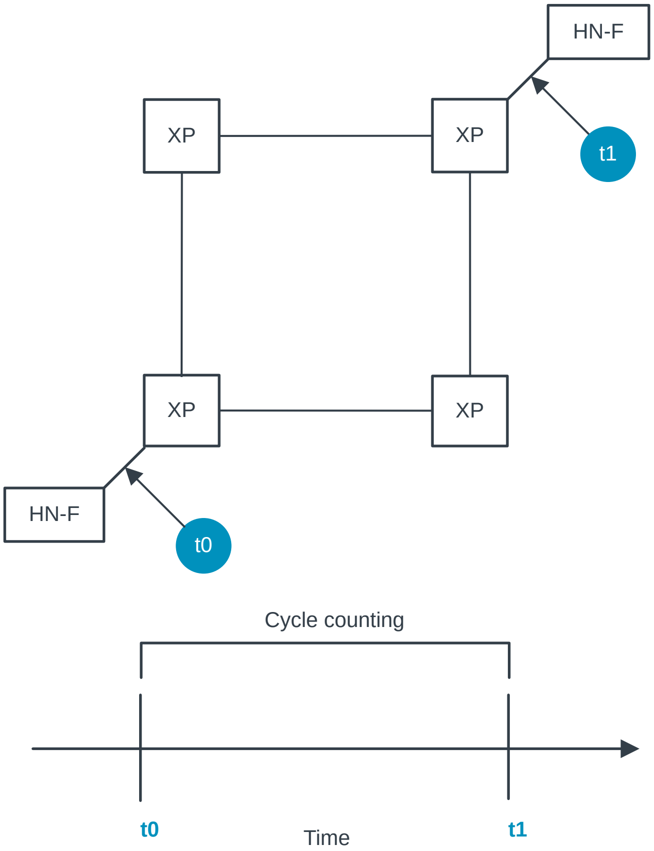
TS# 3-bit encoding of the size of time stamp that is specified as (number of bytes − 1)
CC When set, indicates that a 2B cycle count is included in the packet after the payload

7.1.5.4 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 7-5: Cycle counting



The cycle counter payload is 2B and 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.

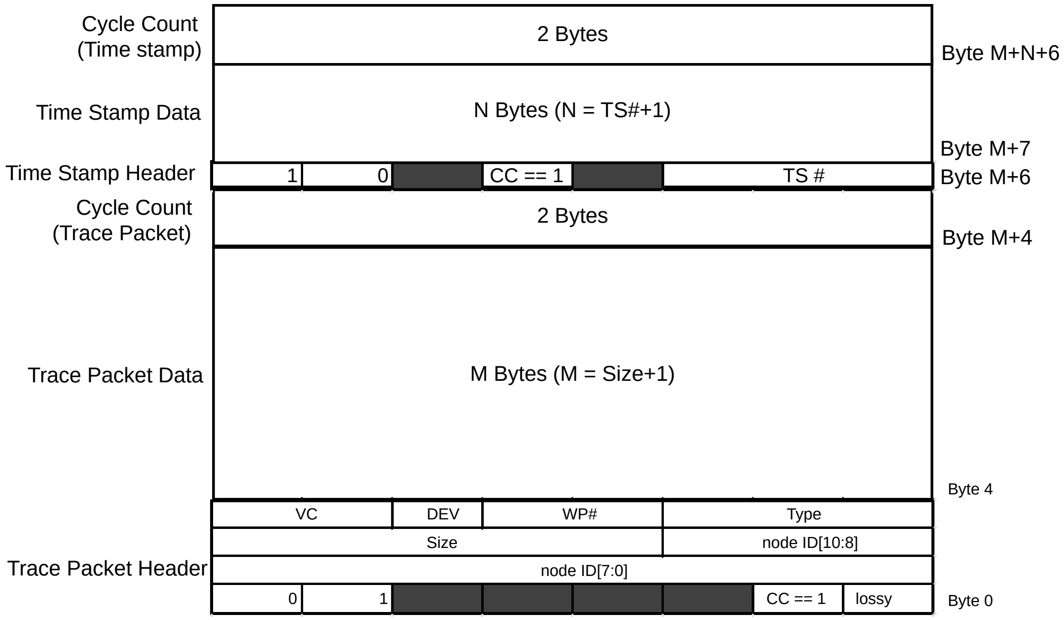
7.1.5.5 Trace stream example

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 7-6: Trace stream



7.2 DT usage examples

To help you use the CMN-650 DT features, we describe some example use cases of the DT system and example programming for those use cases.

7.2.1 Flit tracing

CMN-650 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 7.1.1.1 WP match value and mask register on page 905 and 7.1.2.1 Trace data format on page 909.

7.2.1.1 Flit tracing example

CMN-650 can trace individual flits at device interfaces at each XP.

For more information, refer to 5.4.7.1 Program DTM watchpoint on page 846.

This section 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 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_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 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_sel = RN-F2_port
        2. Select flit channel, wp_chn_sel = REQ
        3. Match format group to secondary for Addr match, wp_grp = 1

Note

In combined mode, use WP0 config settings to enable trace generation.

3. To enable trace generation in the WP, set por_dtm_control.trace_tag_enable = 1
4. Set por_dtm_control.dtm_enable = 1

5. Program por_dt_traceid.traceid according to the *Arm® CoreSight™ Architecture Specification v3.0*. The supported range of values is `0x01-0x6F`
6. Program por_dt_dtc_ctl to enable tracing, dt_en = 1

7.2.2 Trace tag

CMN-650 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.

7.2.2.1 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.

7.2.2.2 Trace tag propagation

All CMN-650 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.

7.2.2.3 Trace tag trace packet generation

When a watchpoint is enabled for trace packet generation through wp_pkt_gen, there are several interactions. Any flit with the TRACETAG field asserted on the channel (which wp_chn_sel selects), for the device (which wp_dev_sel has selected), generates a trace packet type which wp_pkt_type selects.

This trace packet is generated whenever TRACETAG is asserted in a flit; a watchpoint match, determined by wp_val and wp_mask, is not required.

7.2.2.4 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 5.4.7.1 Program DTM watchpoint on page 846.

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_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.Addr:
    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_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

Note

In combined mode, use WP0 config settings to enable trace generation.

3. To enable trace tag generation in the WP, set dtm_control.trace_tag_enable = 1
4. Set dtm_control.dtm_enable = 1
5. Program por_dt_traceid.traceid according to the *Arm® CoreSight™ Architecture Specification v3.0*. The supported range of values is `0x01-0x6F`.
6. Program por_dt_dtc_ctl to enable tracing, dt_en = 1

7.2.3 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.

7.2.4 Cross trigger

CMN-650 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 XP or different XPs) are set up to generate a cross trigger on specific events to the DTC. The DTC is programmed to trigger the DTMs in step 1.

7.2.4.1 Cross trigger example programming

CMN-650 can trigger DTMs based on specific events occurring elsewhere in the system.

For more information, refer to 5.4.7.1 Program DTM watchpoint on page 846.

This example uses trace DAT flits corresponding to a ReadShared transaction to address-X that originated at RN-F 2. There have also been 10 WriteNoSnoops uploaded to the HN-D.

1. Set WP or WPs at all DAT download ports to generate DAT flit traces for ReadShare transactions from RN-F 2 to address-X.
   a. Program WP2 and WP3, which are at the DAT download ports, to trace DAT flits:
      1. Set por_dtm_wp2_val/mask and por_dtm_wp3_val/mask registers to match on SRCID = RN-F 2, opcode = ReadShared, and address = X.
      2. Set por_dtm_wp2_config and por_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.
   b. Enable the WP by setting the dtm_enable bit of the por_dtm_control register to 1.

2. Set up WP at HN-D upload port to monitor WriteNoSnoop flits.
    a. Program WP0 (upload WP) to monitor and enable cross trigger REQ. Opcode as follows:
        1. Set por_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).
    b. Enable WP.
        - Set the dtm_enable field of the por_dtm_control register = 1.
3. Set up counter in DTC to count ten trigger events from step 3.
    - Program por_dt_dtc_ctl as follows:
        a. Set cross trigger count (cross_trigger_count = 10).
        b. Enable waiting for HN-D WP trigger event (dt_wait_for_trigger = 1).
        c. Enable DTC (dt_en = 1).

7.2.4.2 Sample profile

CMN-650 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 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 count down cycle begins.

There is only one set of por_dtm_pmsicr and por_dtm_pmsirr registers per XP, as only one outstanding transaction is expected. Por_dtm_pmsicr is 24 bits, and the lower 8 bits of por_dtm_pmsirr are zero.

In general, Secure transactions are allowed to be tagged and traced with the secure_debug_disable field of the por_dt_secure_access register. When this bit is set, Secure transactions are not traced.

7.3 PMU system overview

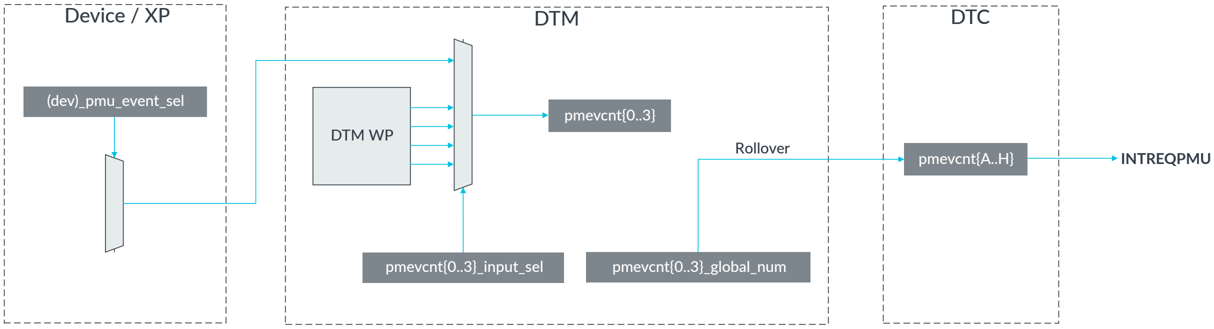
CMN-650 includes *Performance Monitoring Unit* (PMU) capabilities.

The PMU offers the following features:

- Local and global performance counters with shadow registers
- PMU snapshot across all internal CMN-650 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 7-7: PMU local and global performance counters



For the various PMU programming sequences, see 5.4.8 PMU system programming on page 847.

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 either:

- A request of snapshot through input pin **PMUSNAPSHOTREQ**
- A write into the ss_req field of the por_dt_pmsrr 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.

7.4 CXLA PMU system

The CXLA PMU system is present across multiple nodes in the CXG. There are certain details that you must be aware of for the system to function correctly.

The CXLA PMU system has two components: the PMU controller, por_cxg_pmu_ctl, and the PMU counter, por_cxla_pmu.

For the CXLA PMU system to function correctly, the power status of CXLA must be communicated to both DTC and the local CXLA PMU controller. Set the por_cxg_ra_cfg_ctl.en_cxla_pmucmd_prop register field to 1 if CXLA is in the powered up state, and CXLA PMU is functioning. The number of

CXLAs that are powered up is added to the number of DTMs to specify the total number of PMU counters.

Note

The total number of XP and CXRH DTMs and PMUs in a single DTC domain must not be greater than 63.

Note

The PMU events in CXLA are captured in the por_cxla_pmevcnt registers and HA/RA PMU events are captured in the XP por_dtm_pmevcnt register, in the same way as all other devices.

7.5 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.

8 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.

8.1 Performance optimization guidelines

There are some restrictions when optimizing CMN-650.

To obtain maximum performance from CMN-650, the system integrator must be aware of the following information:

RN-I

When request ordering is not required, transaction requests must be dispatched with non-overlapping IDs to ensure optimal bandwidth operation. Large Burst transactions, in other words transactions larger than 64B, must be split into 64B or smaller Burst transactions. 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 example, multiple (partial) WriteUnique transactions must be combined into a single WriteUnique or a single WriteLineUnique transaction. In the resultant transaction, all bytes in the cache line must be 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 in response to read requests from AR channels. Disabling of read data interleaving does not apply to atomic operations. For example, the **RDATA** data from an atomic operation sent on the AW channel can interleave with **RDATA** data from an AR channel request.

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-650 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.

8.1.1 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.

8.2 About the Performance Monitoring Unit

CMN-650 provides access to various performance events. Some of these events are unique to and originate in a specific CMN-650 component. Some 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 5.4.8.1 Set up PMU counters on page 847.

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 7 Debug trace and PMU on page 902.

8.2.1 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.

8.3 HN-F performance events

The HN-F performance analysis counters are used to monitor cache behavior.

For a particular cache, the cache miss or hit rate is used to measure the capacity of the cache, and the location for certain applications. To measure the cache miss rate, the performance monitor counters count the number of instances of cache accesses and cache misses.

8.3.1 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.

Note

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-650 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 8-1: Cache miss rate

$$\text{Cache miss rate (\%)} = \frac{\text{Total cache misses}}{\text{Total cache accesses}} \text{ x } 100$$

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 WriteBack 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, WriteBack, and Evictions that are allocated in SLC cache, are counted towards this event.

8.3.2 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 8-2: HN-F message retry rate

$$\text{HN-F message retry rate (\%)} = \frac{\text{HN-F total messages retried}}{\text{HN-F total messages received}} \text{x } 100$$

8.3.3 SF events

There are three snoop events that can be counted.

The following sections describe the SF performance events.

SF miss rate

This event measures the amount of memory controller traffic that is generated. It can also be used to measure the efficiency of the SF.

PMU_HN_SF_HIT_EVENT Measures the number of SF hits.

Calculate the SF hit rate as follows:

Figure 8-3: 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 8-4: 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.

8.3.4 System-wide events

The *Memory Controller* (MC) request retries determine whether the MC 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 MC.

PMU_HN_MC_REQS_EVENT Total number of requests that are sent to the MC.

Calculate the retry rate for requests to the MC as follows:

Figure 8-5: MC message retry rate

$$\text{MC message retry rate (\%)} = \frac{\text{MC total messages retried}}{\text{MC total messages received}} \text{ x } 100$$

8.3.5 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, this event counts these four snoops as one since all four snoops are sent to the same cluster.

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.

8.3.6 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.

8.3.7 HN-F PMU event summary

The following table shows a summary of the HN-F PMU events.

Table 8-1: 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 due to 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 due to SF back invalidation.
26	PMU_HN_SFBI_BRD_SNP_SENT_EVENT	Counts number of times multicast snoops were sent due to SF back invalidation.
27	PMU_HN_SNP_SENT_UNTRK_EVENT	Counts number of times snoops were sent due to untracked RN-Fs.

Number	Name	Description
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_SNP_SENT_CLUSTER_EVENT	Counts number of snoops sent to clusters excluding individual snoops within a cluster.
33	PMU_HN_SF_IMPRECISE_EVICT_EVENT	Counts number of times an evict operation was dropped due to SF clustering.
34	PMU_HN_SF_EVICT_SHARED_LINE_EVENT	Counts number of times a shared line was evicted from SF.

8.4 RN-I performance events

External devices connect at an RN-I bridge.

8.4.1 Bandwidth at RN-I bridges

External devices connect at an RN-I bridge.

The following events measure bandwidth at the RN-I bridges:

- 8.4.1.1 Requested read bandwidth at RN-I bridges on page 933.
- 8.4.1.2 Actual read bandwidth on interconnect on page 934.
- 8.4.1.3 Write bandwidth at RN-I bridges on page 934.

8.4.1.1 Requested read bandwidth at RN-I bridges

External devices connect to CMN-650 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 8-6: Read bandwidth calculation

$$\text{Read bandwidth} = \frac{\text{Number RDataBeats\_Port}n \text{ x AXIDataBeatSize}}{\text{Cycles}} \text{ x Frequency}$$

Where AXIDataBeatSize is the number of bytes for each AXI beat. 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.

8.4.1.2 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.

This event includes the replayed requests because of the read data buffer decoupled scheme.

Calculate the actual read bandwidth as follows:

Figure 8-7: Actual read bandwidth

$$\text{Actual read bandwidth} = \frac{\text{RXDATFLITV x DataFlitSize}}{\text{Cycles}} \text{x Frequency}$$

8.4.1.3 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 8-8: Actual write bandwidth

$$\text{Actual write bandwidth} = \frac{\text{TXDATFLITV x DataFlitSize}}{\text{Cycles}} \text{x Frequency}$$

8.4.2 Bottleneck analysis at RN-I bridges

CMN-650 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 masters.
- The number of read request replays, because of decoupling of the read request buffers and read data buffers in the RN-I system.

8.4.2.1 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 8-9: Retry rate

$$\text{Retry rate} = \frac{\text{TXREQFLITV\_RETRIED}}{\text{TXREQFLITV\_TOTAL}}$$

8.4.2.2 Read and write delays at RN-I bridges

To monitor the delays for reads and writes, CMN-650 lets you monitor how full the read and write trackers are in the RN-I bridges.

When one of the trackers is full, the bridge cannot accept new requests from the AXI master. This condition delays the I/O devices that connect to the AXI master.

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 the following situations:

- The read tracker of a specific RN-I bridge is full but the effective read bandwidth from the bridge is not close to the maximum expected. In this case, the interconnect cannot keep up with the read traffic from the specific device.
- The bandwidth is close to maximum, the I/O device can send requests to the maximum of its port bandwidth. In this case, 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 masters.

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.

Note

For CMN-650, 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 according to:

- A hash of the **ARID** value of the request
- Which of the three ACE-Lite slave interfaces receives the request

Therefore, in these configurations, the maximum number of outstanding same-**ARID** requests from the same ACE-Lite slave interface is 64.

The *RRT_OCCUPANCY* event covers the total occupancy of all read tracker slices.

8.4.3 RN-I PMU event summary

There are 16 RN-I PMU events.

The following table shows a summary of the RN-I PMU events.

Table 8-2: 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.
8	PMU_RNI_RRT_OCCUPANCY	All entries in the 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.

Number	Name	Description
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.

8.5 SBSX performance events

This section contains SBSX performance event information.

8.5.1 Bandwidth at SBSX bridges

This section contains SBSX bridge bandwidth information.

The following events are used to measure bandwidth at the SBSX bridges:

- 8.5.1.1 Read bandwidth on interconnect at SBSX bridges on page 937.
- 8.5.1.2 Write bandwidth at SBSX bridges on page 938.
- 8.5.1.3 Total requested bandwidth at SBSX bridges on page 938.

8.5.1.1 Read bandwidth on interconnect at SBSX bridges

This section 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 8-10: 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.

8.5.1.2 Write bandwidth at SBSX bridges

This section 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 8-11: Actual write bandwidth

$$\text{Actual write bandwidth} = \frac{\text{PMU\_SBSX\_TXDAT} \ \text{x DataFlitSize}}{\text{Cycles}} \text{ x Frequency}$$

Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

8.5.1.3 Total requested bandwidth at SBSX bridges

This section 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 8-12: 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.

8.5.2 Bottleneck analysis at SBSX bridges

This section contains information on bottleneck analysis at SBSX bridges.

CMN-650 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-650 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 due to backpressures on AXI or CHI interface.

The following events are used to measure bottlenecks at the SBSX bridges:

- 8.5 SBSX performance events on page 937.

8.5.2.1 Request retry rate at SBSX bridges

This section 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

8.5.2.2 Delays at SBSX bridges due to backpressure

To analyze the delays in SBSX bridges, CMN-650 enables you to monitor the source of backpressure.

SBSX might have requests that are ready to be sent to the downstream AXI or ACE-Lite device. However, it cannot send them due to 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 8-3: AXI or ACE-Lite downstream monitoring events

Events	Description
ARVALID_NO_ARREADY	Number of cycles the SBSX bridge is stalled due to backpressure on AR channel.
AWVALID_NO_AWREADY	Number of cycles the SBSX bridge is stalled due to backpressure on AW channel.
WVALID_NO_WREADY	Number of cycles the SBSX bridge is stalled due to 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 8-4: 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.

8.5.2.3 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 8-5: 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

8.5.3 SBSX PMU event summary

This section contains SBSX PMU event summary information.

For more information, see 5.3.11.18 por_sbsx_pmu_event_sel on page 590.

8.6 HN-I performance events

This section contains HN-I performance event information.

8.6.1 Bandwidth at HN-I bridges

This section contains HN-I bridge bandwidth information.

The following events are used to measure bandwidth at the HN-I bridges:

- 8.6.1.1 Read bandwidth on interconnect at HN-I bridges on page 941.
- 8.6.1.2 Write bandwidth at HN-I bridges on page 942.
- 8.6.1.3 Total requested bandwidth at HN-I bridges on page 942.

8.6.1.1 Read bandwidth on interconnect at HN-I bridges

This section 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 8-13: 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.

8.6.1.2 Write bandwidth at HN-I bridges

This section 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 8-14: 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.

8.6.1.3 Total requested bandwidth at HN-I bridges

This section 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 8-15: Total requested bandwidth

$$\text{Total requested bandwidth} = \frac{\text{PMU\_HNI\_TXREQ} \ \text{x DataFlitSize}}{\text{Cycles}} \text{ x Frequency}$$

Note

This event is tracked in the DTM watchpoint in the XP where the component is located.

8.6.2 Bottleneck analysis at HN-I bridges

This section 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-650 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-650 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 due to ordering requirements.
- The number of cycles the bridge is forced to stall due to backpressures.

The following events are used to measure bottlenecks at the HN-I bridges:

- 8.6 HN-I performance events on page 941.

8.6.2.1 Request retry rate at HN-I bridges

This section 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

8.6.2.2 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 8-6: 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.

8.6.2.3 Delays at HN-I bridges because of backpressure

To analyze the delays in HN-I bridges, CMN-650 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. However, it 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 8-7: AXI and ACE-Lite downstream events monitor information

Events	Description
ARVALID_NO_ARREADY	Number of cycles the HN-I bridge is stalled due to backpressure on AR channel.
AWVALID_NO_AWREADY	Number of cycles the HN-I bridge is stalled due to backpressure on AW channel.
WVALID_NO_WREADY	Number of cycles the HN-I bridge is stalled due to 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 AXI or ACE-Lite downstream:

Table 8-8: AXI and ACE-Lite 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 8-9: CHI events monitor information

Events	Description
TXDATFLITV_NO_LINKCRD	Number of cycles the **TXDAT** flit in HN-I bridge is waiting for link credits.

8.6.2.4 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 8-10: 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

8.6.3 HN-I PMU event summary

This section contains HN-I PMU event summary information.

Refer to 5.3.3.21 por_hni_pmu_event_sel on page 320 for more information.

8.6.4 HN-P PMU events

HN-P has dedicated RRTs, RDTs, and WDBs for requests from RN-Is and RN-Ds connected to PCIe masters. There are associated PMU events for these resources.

For more information about the associated events, see 5.3.3.22 por_hnp_pmu_event_sel on page 322.

Reads from PCIe RN-Is and RN-Ds do not support Bypasstracetag.

8.7 DN performance events

This section contains DN performance event information.

The following table shows a summary of the DN PMU events.

Table 8-11: 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 due to lack of credits
11	DVM tracker full counter
12	DVM tracker occupancy counter

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 8-12: Field values for pmu_occup1_id

pmu_occup1_id values	Description
`0b0000`	All
`0b0001`	DVM ops
`0b0010`	DVM syncs

Note

In HN-D, 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. The por_dn_pmu_event_sel register outputs on corresponding TXPMU output only if por_hni_pmu_event_sel bit[5], bit[13], bit[21], and bit[29] are set to 0. Otherwise the value on the HN-I PMU is available.

The por_hnp_pmu_event_sel register only outputs on the corresponding TXPMU output if por_hni_pmu_event_sel bit[5], bit[13], bit[21], and bit[29] are set to 0. Otherwise, the value on the HN-I PMU is available.

8.8 XP PMU event summary

This section contains XP PMU event summary information.

Each of the XP PMU events is associated with:

- One of six XP ports - east, west, north, south, device port0 or device port1.
- One of six CHI channels - REQ, RSP, SNP, DAT, RSP2, or DAT2. RSP2 and DAT2 are valid when dual DAT and RSP mode is enabled.

You can specify up to four XP PMU events using the por_mxp_pmu_event_sel register. For more information about this register, see 5.3.14.17 por_mxp_pmu_event_sel on page 668.

The following table shows a summary of the XP PMU events.

Table 8-13: 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.

8.9 CXG performance events

This section contains CXG performance event information.

The XP considers CXRH as a single node when selecting PMU events. The PMU events for CXRA and CXHA share the four possible PMU counters for that port in the XP. Therefore a total of four events can be counted simultaneously from CXRA+CXHA. The CXHA events have "priority" over the CXRA events. If an event selection (0, 1, 2, 3) in por_cxg_ha_pmu_event_sel is nonzero (is not CXHA_PMU_EVENT_NULL), then the event selected in por_cxg_ha_pmu_event_sel is sent to the XP. If an event selection (0, 1, 2, 3) in por_cxg_ha_pmu_event_sel is zero (is CXHA_PMU_EVENT_NULL), then the event selected in por_cxg_ra_pmu_event_sel is sent to the XP.

The CXLA contains its own PMU event counters. As with XP PMU counters, you can pair CXLA PMU event counters with the global counters in a DTC. Configure the CXLA PMU event counters in por_cxla_pmuevcnt.

Note

To enable CXLA PMU functionality, you must configure por_cxla_pmu_config.pmu_en and por_cxg_ra_cfg_ctl.en_cxla_pmucmd_prop to 1'b1. Also configure the usual DT and PMU enables.

8.10 Occupancy and lifetime measurement using PMU events

CMN-650 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-650 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 8-16: Average occupancy

$$\text{Average Occupancy (entries)} = \frac{\text{PMU\_OCCUPANCY\_EVENT} \ll 12}{\text{PMU\_CYCLE\_COUNTER}}$$

For example, for RN-I RRT average occupancy, the formula is:

Figure 8-17: Average RRT occupancy

$$\text{Average RRT Occupancy (entries)} = \frac{\text{PMU\_RNI\_RRT\_OCCUPANCY\_EVENT} \ll 12}{\text{PMU\_CYCLE\_COUNTER}}$$

Lifetime measurement

If a tracker supports lifetime event, the formula to measure the lifetime is:

Figure 8-18: Average lifetime

$$\text{Average Lifetime (cycles)} = \frac{\text{PMU\_OCCUPANCY\_EVENT} \ll 12}{\text{PMU\_NUM\_TRACKER\_ALLOCATIONS}}$$

For example, for RN-I RRT average lifetime, the formula is:

Figure 8-19: Average RRT lifetime

$$\text{Average Lifetime (cycles)} = \frac{\text{PMU\_RNI\_RRT\_OCCUPANCY} \ll 12}{\text{PMU\_RNI\_RRTALLOC}}$$

HN-F supports collecting occupancy according to the request types. The following table describes the opcode filtering types that are supported.

Table 8-14: 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 you enable filtering, pmu_occup1_id must return to the default value to collect occupancy for all request types.

8.11 DEVEVENT

CMN-650 HN-Fs support device-specific events that are collectively called DEVEVENT. These events are sent along with the completion of a transaction.

The completion of a transaction can be a data response (DAT) or completion response (RSP). These events contain information regarding the transaction encountering an SLC hit or miss. The events also include information about snoops sent to resolve coherency actions. You can measure these events using watchpoints on the XP that the RN-F is connected to. Refer to 7.1.1 DTM watchpoint on page 904 for watchpoint usage.

The following table describes the DEVEVENT encodings from HN-F.

Table 8-15: 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
2'b10	Line missed in SLC and broadcast snoops sent
2'b11	Line hit in SLC and no snoops sent

Other CMN-650 device responses have the default 2’b00 encoding as the DEVEVENT value.

Appendix A Protocol feature compliance

This appendix describes the various features that CMN-650 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-650 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-650 nodes with AXI or ACE-Lite interfaces support.

Table A-1: AXI and ACE-Lite feature support

AXI or ACE-Lite property	Support		
	RN-I	HN-I	SBSX
Wakeup_Signals	Y	Y	Y
Check_Type	Y	Y	Y
Poison	Y	Y	Y
Trace_Signals	Y	Y	Y
Unique_ID_Support	Y	Y	Y
QoS_Accept	N	N	N
Loopback_Signals	Y	N	N
Untranslated_Transactions	N	N	N
NSAccess_Identifiers	Y	N	Y
CMO_On_Read	Y	N	Y
CMO_On_Write	Y	N	Y
Persist_CMO	Y	N	Y
Write_Plus_CMO	N	N	N
DVM_v8 and DVM_v8.1	Y	N	N
DVM_v8.4	N	N	N
Coherency_Connection_Signals	Y	N	N
MPAM_Support	Y	Y	Y
Read_Interleaving_Disabled	Y	Y	Y
Read_Data_Chunking	Y	N	N
Cache_Stash_Transactions	Y	N	N
Atomic_Transactions	Y	N	N
DeAllocation_Transactions	Y[1]	N	N

[1] ReadOnceMakeInvalid or ReadOnceCleanInvalid

AXI or ACE-Lite property	Support		
	RN-I	HN-I	SBSX
WriteEvict_Transaction	N	N	N
Barrier_Transactions	N	N	N
Ordered_Write_Observation	Y	N	N
DVM_On_Read	Y (RN-D)	N	N
DVM_On_Snoop	Y (RN-D)	N	N
Max_Transaction_Size	Any	64B	64B
Fixed_Burst	Y	N	N
Exclusive_Access	Y	N	N
Shareable_Transactions?	Y	N	N
Prefetch_Transaction	N	N	N

A.2 CHI feature support

CHI provides various optional features through interface properties. CMN-650 supports some of these properties.

The following table shows the CHI properties that CMN-650 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	-
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.
CleanSharedPersistSep_Request	Y	-
MPAM_Support	Y	-

A.3 CXS property support

This section provides information on CXS properties and their relevant support.

The following table contains the supported CXS properties.

Table A-3: CXS property support

CXS property	Support	Comments
TX and RX CXSDATAFLITWIDTH	256/512	CXSDATAFLITWIDTH values are always the same for CXS TX and RX.
TX and RX CXSMAXPKTPERFLIT	2/4	If TX/RX CXSDATAFLITWIDTH = 512, the value of TX/RX CXSMAXPKTPERFLIT = 4.
TX CXSCONTINUOUSDATA	True	-
RX CXSCONTINUOUSDATA	False	-
TX and RX CXSERRORFULLPKT	True	-
TX and RX CXSDATACHECK	None	-
TX and RX CXSREPLICATION	None	-

A.4 CCIX property support

This section provides CCIX information for CML support.

The following table contains CCIX property settings for CML.

Table A-4: CCIX property settings for CML

Property	Permitted values	Support
NoCompAck	True, false	False
PartialCacheStates	True, false	False
CacheLineSize	64B, 128B	64B
AddrWidth	48b, 52b, 56b, 60b, 64b	48b/52b
PktHeader	Compatible, Optimized	Both
MaxPacketSize	128B, 256B, 512B	All
NoMessagePack	True, false	Both (true is the default)

Note

The following CCIX features are not supported:

- Concurrent memory expansion.
- Snoop chaining outbound.
- CCIX snoop multicast (inbound and outbound).
- Snoop broadcast outbound.

A.5 CHI feature support for CML

This section provides CHI information for *Coherent Multichip Link* (CML) support.

The following table contains CHI support for CML settings.

Table A-5: CHI support for CML

CHI feature	CML support		Comments
	Local	Remote	
Coherency	Yes	Yes	-
Ordering	Yes	Yes	-
Atomics	Yes	Yes	-
Exclusive Accesses	Yes	SMP mode only	Remote support. Only RN-F exclusives are supported and can only target HN-F. In non-SMP mode, any read exclusive access is downgraded to a non-exclusive read and write exclusive is terminated at CXRA and Non-data Error response is sent.
Cache Stashing	Yes	No	-
DVM operations	Yes	SMP mode only	-
Error handling			
- Response Error	Yes	Yes	-
- Data Check	Yes	No	-
- Poison	Yes	Yes	Mandatory for CMN-650
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 CCIX gateway block
Direct Memory Transfer (DMT)	Yes	No	Local support includes local SN-F sending data directly to CCIX gateway block
I/O Deallocation transactions	Yes	Yes	-
CleanSharedPersist CMO	Yes	Yes	-
CleanSharedPersistSep CMO	Yes	No	-
Prefetch Target	Yes	No	Remote support. This request is intended to target SN and therefore is not expected at CXRA.
Trace Tag	Yes	SMP mode only	-
System Coherency Interface (**SYSCOREQ** and **SYCOACK**)	Yes	Yes (using s/w bits)	-

CHI feature	CML support		Comments
	Local	Remote	
Partial Cache State	Yes	No	CXRA, inside CML block, does not accept the following requests and responses: - WriteBackPtl - SnpRespDataPtl*
Streaming and optimized Streaming of Ordered WriteUniques	Yes	No	WriteUnique* request with Request Order and ExpCompAck attributes set, RN must send the CompAck when it receives a Comp response for that write request. The RN must not create any dependencies on other outstanding writes.
MPAM	Yes	SMP mode only	-

Appendix B Signal descriptions

This appendix describes the external I/O signals that CMN-650 implements for connection to other hardware in the system.

B.1 About the signal descriptions

CMN-650 signals are composed of a base name along with identifiers that indicate unique product configuration.

Because there are multiple identical interfaces in CMN-650, 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-650 signals are active-HIGH.

B.2 Clock and reset signals

CMN-650 has global clock and reset signals for controlling the clock and reset functionality.

Depending on your CMN-650 clock domain configuration, the interconnect might have a single global clock signal or multiple global clock signals.

Signal definitions

Table B-1: CMN-650 clock and reset signals

Signal	Direction	Description	Connection information
GCLK0	Input	Primary CMN-650 clock input. This clock signal is always present.	Connect to global clock for CMN-650.
GCLK1	Input	Primary CMN-650 clock input. This clock is only present if you divide the mesh into four asynchronous clock domains.	Connect to global clock for CMN-650.
GCLK2	Input	Primary CMN-650 clock input. This clock is only present if you divide the mesh into four asynchronous clock domains.	Connect to global clock for CMN-650.
GCLK3	Input	Primary CMN-650 clock input. This clock is only present if you divide the mesh into four asynchronous clock domains.	Connect to global clock for CMN-650.
nSRESET	Input	CMN-650 reset, active-LOW	Connect to global reset for CMN-650.

B.3 CHI interface signals

CMN-650 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-650 interfaces use **RootName** within a more fully specified signal name as follows:

- CMN-650 interface signal name == **RootName_NID#**, where # is the node ID corresponding to the specific interface.

B.3.1 Per-device interface definition

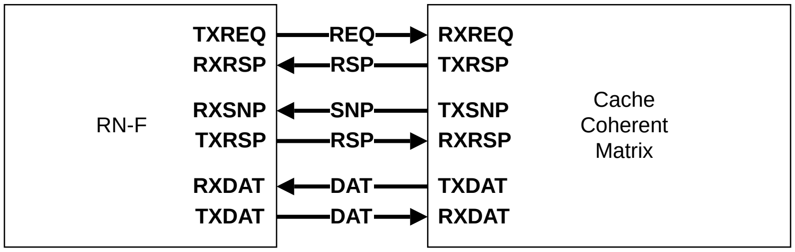
Each CHI device included in a CMN-650 system has distinct functionality, and the requirements and configuration of its respective CHI interfaces differ.

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

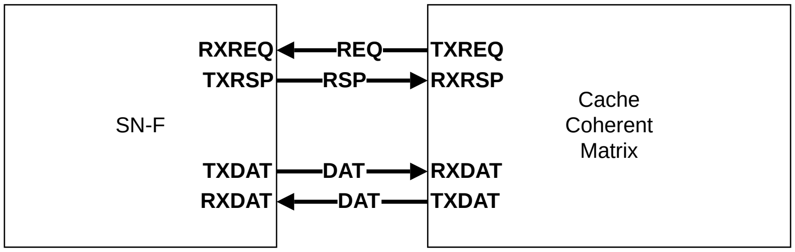


It also has two data channels, one in each direction, for data transfers. CMN-650 receives request messages from the RN-F and sends responses to it. In addition, CMN-650 sends snoop messages to the RN-F and receives snoop response messages.

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-650 and returns response messages.

B.3.2 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.

Note

Connection of CHI interfaces between two devices requires cross-coupling of the **TX*** and **RX*** signals between the two devices, as required by the CHI architecture.

Signal definitions

Table B-2: 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][2]	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][2]	Output	Transmit Response Flit	Connect to **RXRSPFLIT** of the corresponding CHI device, if populated.

[2] The value of *n* is configuration-dependent.

Signal	Direction	Description	Connection information
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][2]	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][2]	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.
RXREQFLIT[n:0][2]	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][2]	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][2]	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][2]	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.3.3 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-3: 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.
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-650 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.3.4 DAT.RSVDC subfields

The structure of the RSVDC field and its subfields in the CMN-650 DAT channel depends on the value of specific configuration parameters.

The following table shows the DAT.RSVDC subfields and information about how your configuration affects these fields.

Table B-4: DAT.RSVDC subfields

Subfield	Type	Description	Connection information
BASE[BASE_WIDTH-1:0]	Input/ output	Base	*BASE_WIDTH* = 4

Subfield	Type	Description	Connection information
METADATA[METADATA_WIDTH-1:0]	Input/ output	Metadata	The *RSVDC_METADATA_WIDTH* parameter controls the size of *METADA\TA_WIDTH*. The **METADATA** subfield is not present if *META_DATA_EN* = 0.

To create the DAT.RSVDC field, the subfields in the preceding table are fully packed in the same order as shown. The BASE subfield represents the LSB side of the field.

B.4 ACE-Lite and AXI interface signals

CMN-650 interfaces use **RootName** as the signal name within a more fully specified convention.

Note

All signal names in this section consist of a root name, **RootName**. CMN-650 interfaces use **RootName** within a more fully specified signal name as follows:

CMN-650 ACE-Lite and AXI interface signal name == **RootName_[S|M]<#a>_NID#b**, where:

S|M Defines either a slave or master interface.

#$_a$ Defines an optional interface identifier for a node that can support multiple AMBA interfaces.

#$_b$ Defines the node ID corresponding to the specific interface.

Multi-bit signals append the bit-range identifier included in the **RootName** to the end of the full signal name.

B.4.1 ACE-Lite-without-DVM slave interface signals

This interface is present as the ACE-Lite-without-DVM slave port for an RN-I bridge.

Note

For more information about the ACE-Lite-with-DVM slave interface for an RN-D bridge, see B.4.2 ACE-Lite-with-DVM slave interface signals on page 964.

Signal definitions

Table B-5: ACE-Lite-without-DVM slave 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.
AWAKEUP_S	Input	Indication that the master is starting a transaction that is being sent to the interconnect	Connect to corresponding master device, if populated, otherwise tie LOW.

Signal	Direction	Description	Connection information
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.
AWREADY_S	Output	Write address ready	Connect to corresponding master device, if populated.
AWVALID_S	Input	Write address valid	Connect to corresponding master device, if populated, otherwise tie LOW.
AWID_S[10:0]	Input	Write address ID	
AWADDR_S[n:0][3]	Input	Write address	
AWLEN_S[7:0]	Input	Write burst length	
AWSIZE_S[2:0]	Input	Write burst size	
AWBURST_S[1:0]	Input	Write burst type	
AWLOCK_S	Input	Write lock type	
AWCACHE_S[3:0]	Input	Write memory type	
AWUSER_S[n:0]	Input. Where n = (REQ_RSVDC_WIDTH-1).	User-defined signal	
AWPROT_S[2:0]	Input	Write protection type	
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 master 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	
AWSTASHNIDEN_S	Input	When asserted, indicates that the **AWSTASHNID** signal is valid and should be used	
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	
AWSTASHLPIDEN_S	Input	When asserted, indicates that the **AWSTASHLPID** signal is enabled and should be used	
AWTRACE_S	Input	Trace signal that is associated with the AW channel	
AWLOOP_S[1:0]	Input	Loopback signal	

[3] The value of *n* is configuration-dependent.

Signal	Direction	Description	Connection information
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 master device, if populated, otherwise tie LOW.
AWIDUNQ_S	Input	Unique ID indicator signal	
AWNSAID_S[3:0]	Input	Non-secure Access Identifier signal	
AWCMO_S[1:0]	Input	Write address channel CMO indicator	
WREADY_S	Output	Write data ready	Connect to corresponding master device, if populated.
WVALID_S	Input	Write data valid	Connect to corresponding master device, if populated, otherwise tie LOW.
WDATA_S[n:0][3]	Input	Write data	
WSTRB_S[d:0] [4]	Input	Write byte lane strobes	
WLAST_S	Input	Write data last transfer indication	
WUSER_S[u:0][5]	Input	**WUSER_S[0]** is **WDATACHK** valid signal. If *`META_DATA_EN`* = 1, tie **WUSER_S[u:1]** to 0.	
WTRACE_S	Input	Trace signal	
WPOISON_S[p:0][6]	Input	Poison signal	
WDATACHK_S[d:0][4]	Input	Data check signal	
BREADY_S	Input	Write response ready	Connect to corresponding master device, if populated, otherwise tie LOW.
BVALID_S	Output	Write response valid	Connect to corresponding master device, if populated.
BID_S[10:0]	Output	Write response ID	
BRESP_S[1:0]	Output	Write response	
BUSER_S[3:0]	Output	User response signal	
BTRACE_S	Output	Trace signal	Connect to corresponding master device, if populated.
BLOOP_S[1:0]	Output.	Loopback signal	
BIDUNQ_S	Output	Unique ID indicator signal	

[4] The value of `d` = (((n + 1) / 8) - 1).

[5] 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.

[6] The value of `p` = ceil(*`DATA_WIDTH`* / 64) - 1.

Signal	Direction	Description	Connection information
ARREADY_S	Output	Read address ready	Connect to corresponding master device, if populated.
ARVALID_S	Input	Read address valid	Connect to corresponding master device, if populated, otherwise tie LOW.
ARID_S[10:0]	Input	Read address ID	
ARADDR_S[n:0][3]	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. Where n = (REQ_RSVDC_WIDTH-1).	User-defined signal	
ARPROT_S[2:0]	Input	Read protection type	
ARQOS_S[3:0]	Input	Read QoS value	
ARSNOOP_S[3:0]	Input	Read transaction type	
ARDOMAIN_S[1:0]	Input	Read Shareability domain	
ARTRACE_S	Input	Trace signal	Connect to corresponding master device, if populated, otherwise tie LOW.
ARLOOP_S[1:0]	Input	Loopback signal	
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`	
ARIDUNQ_S	Input	Unique ID indicator signal	
ARCHUNKEN_S	Input	Chunk enable signal. If asserted, read data for this transaction can be returned out of order, in 128-bit chunks.	
ARNSAID_S[3:0]	Input	Non-secure Access Identifier signal	
RREADY_S	Input	Read data ready	Connect to corresponding master device, if populated, otherwise tie LOW.
RVALID_S	Output	Read data valid	Connect to corresponding master device, if populated.
RID_S[10:0]	Output	Read data ID	
RDATA_S[n:0][3]	Output	Read data	
RRESP_S[1:0]	Output	Read data response	
RLAST_S	Output	Read data last transfer indication	

Signal	Direction	Description	Connection information
RUSER_S[x:0][7]	Output.	**RUSER_S[0]** is **RDATACHK** valid signal	
RTRACE_S	Output	Trace signal	Connect to corresponding master device, if populated.
RPOISON_S[p:0][6]	Output	Poison signal	
RDATACHK_S[d:0][4]	Output	Data check signal	
RLOOP_S[1:0]	Output	Loopback signal	
RIDUNQ_S	Output	Unique ID indicator signal	
RCHUNKV_S	Output	If asserted, **RCHUNKNUM** and **RCHUNKSTRB** are valid for this transfer.	
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.	
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]**	

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, the**WUSER_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.

[7] 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.

B.4.2 ACE-Lite-with-DVM slave interface signals

This interface is present as the ACE-Lite-with-DVM slave port for an RN-D bridge.

Note

For more information about the ACE-Lite-without-DVM slave interface for an RN-I bridge, see B.4.1 ACE-Lite-without-DVM slave interface signals on page 960.

Signal definitions

Table B-6: ACE-Lite-with-DVM slave 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 master (SMMU)	Connect to corresponding master device, if populated.
AWAKEUP_S	Input	Indication that the master is starting a transaction that is being sent to the interconnect	Connect to corresponding master 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 master device. Tie LOW if the master interface is not populated or if the master interface does not have a **SYSCOREQ_S** output port.
SYSCOACK_S	Output	Acknowledge for DVM domain entry or exit	Connect to corresponding master device, if populated.
AWREADY_S	Output	Write address ready	Connect to corresponding master device, if populated.
AWVALID_S	Input	Write address valid	Connect to corresponding master device, if populated, otherwise tie LOW.
AWID_S[10:0]	Input	Write address ID	
AWADDR_S[n:0][8]	Input	Write address	
AWLEN_S[7:0]	Input	Write burst length	
AWSIZE_S[2:0]	Input	Write burst size	
AWBURST_S[1:0]	Input	Write burst type	
AWLOCK_S	Input	Write lock type	
AWCACHE_S[3:0]	Input	Write memory type	
AWUSER_S[n:0]	Input. Where n = (REQ_RSVDC_WIDTH-1).	User-defined signal	
AWPROT_S[2:0]	Input	Write protection type	
AWQOS_S[3:0]	Input	Write *Quality of Service* (QoS) identifier	

[8] The value of *n* is configuration-dependent.

Signal	Direction	Description	Connection information
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 master 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	
AWSTASHNIDEN_S	Input	When asserted, indicates that the **AWSTASHNID** signal is valid and should be used	
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	
AWSTASHLPIDEN_S	Input	When asserted, indicates that the **AWSTASHLPID** signal is enabled and should be used	
AWTRACE_S	Input	Trace signal that is associated with the AW channel	
AWLOOP_S[1:0]	Input	Loopback signal	
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 master device, if populated, otherwise tie LOW.
AWIDUNQ_S	Input	Unique ID indicator signal	
AWNSAID_S[3:0]	Input	Non-secure Access Identifier signal	
AWCMO_S[1:0]	Input	Write address channel CMO indicator	
WREADY_S	Output	Write data ready	Connect to corresponding master device, if populated.
WVALID_S	Input	Write data valid	Connect to corresponding master device, if populated, otherwise tie LOW.
WDATA_S[n:0][8]	Input	Write data	
WSTRB_S[d:0] [9]	Input	Write byte lane strobes	

[9] The value of *d* = (((n + 1) / 8) - 1).

Signal	Direction	Description	Connection information
WLAST_S	Input	Write data last transfer indication	
WUSER_S[u:0][10]	Input	**WUSER_S[0]** is **WDATACHK** valid signal. If *META_DATA_EN* = 1, tie **WUSER_S[u:1]** to 0.	
WTRACE_S	Input	Trace signal	
WPOISON_S[p:0][11]	Input	Poison signal	
WDATACHK_S[d:0][9]	Input	Data check signal	
BREADY_S	Input	Write response ready	Connect to corresponding master device, if populated, otherwise tie LOW.
BVALID_S	Output	Write response valid	Connect to corresponding master device, if populated.
BID_S[10:0]	Output	Write response ID	
BRESP_S[1:0]	Output	Write response	
BUSER_S[3:0]	Output	User response signal	
BTRACE_S	Output	Trace signal	Connect to corresponding master device, if populated.
BLOOP_S[1:0]	Output.	Loopback signal	
BIDUNQ_S	Output	Unique ID indicator signal	
ARREADY_S	Output	Read address ready	Connect to corresponding master device, if populated.
ARVALID_S	Input	Read address valid	Connect to corresponding master device, if populated, otherwise tie LOW.
ARID_S[10:0]	Input	Read address ID	
ARADDR_S[n:0][8]	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. Where n = (REQ_RSVDC_WIDTH-1).	User-defined signal	
ARPROT_S[2:0]	Input	Read protection type	
ARQOS_S[3:0]	Input	Read QoS value	
ARSNOOP_S[3:0]	Input	Read transaction type	
ARDOMAIN_S[1:0]	Input	Read Shareability domain	
ARTRACE_S	Input	Trace signal	Connect to corresponding master device, if populated, otherwise tie LOW.
ARLOOP_S[1:0]	Input	Loopback signal	

[10] 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.

[11] The value of p = ceil(*DATA_WIDTH* / 64) - 1.

Signal	Direction	Description	Connection information
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`	
ARIDUNQ_S	Input	Unique ID indicator signal	
ARCHUNKEN_S	Input	Chunk enable signal. If asserted, read data for this transaction can be returned out of order, in 128-bit chunks.	
ARNSAID_S[3:0]	Input	Non-secure Access Identifier signal	
RREADY_S	Input	Read data ready	Connect to corresponding master device, if populated, otherwise tie LOW.
RVALID_S	Output	Read data valid	Connect to corresponding master device, if populated.
RID_S[10:0]	Output	Read data ID	
RDATA_S[n:0][8]	Output	Read data	
RRESP_S[1:0]	Output	Read data response	
RLAST_S	Output	Read data last transfer indication	
RUSER_S[x:0][12]	Output.	**RUSER_S[0]** is **RDATACHK** valid signal	
RTRACE_S	Output	Trace signal	Connect to corresponding master device, if populated.
RPOISON_S[p:0][11]	Output	Poison signal	
RDATACHK_S[d:0][9]	Output	Data check signal	
RLOOP_S[1:0]	Output	Loopback signal	
RIDUNQ_S	Output	Unique ID indicator signal	
RCHUNKV_S	Output	If asserted, **RCHUNKNUM** and **RCHUNKSTRB** are valid for this transfer.	

[12] 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.

Signal	Direction	Description	Connection information
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.	
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]**	
ACREADY_S	Input	Snoop address ready	Connect to corresponding master device, if populated, otherwise tie LOW.
ACVALID_S	Output	Snoop address valid	Connect to corresponding master device, if populated.
ACADDR_S[n:0][8]	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 master device, if populated.
CRVALID_S	Input	Snoop response valid	Connect to corresponding master device, if populated, otherwise tie LOW.
CRRESP_S[4:0]	Input	Snoop response	
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, the**WUSER_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.4.3 AXI/ACE-Lite master interface signals

HN-I and SBSX have an AXI/ACE-Lite master interface.

Signal definitions

Table B-7: AXI/ACE-Lite master interface signals

Signal	Direction	Description	Connection information
ACLKEN_M	Input	AXI master bus clock enable signal	Connect to clock-enable logic.
AWAKEUP_M	Output	Indicates that CMN-650 is starting an AXI transaction	Connect to corresponding slave device, if populated.
AWREADY_M	Input	Write address ready	Connect to corresponding slave device, if populated, otherwise tie LOW.
AWVALID_M	Output	Write address valid	Connect to corresponding slave device, if populated.
AWID_M[x:0][13]	Output	Write address ID	
AWADDR_M[n:0][14]	Output	Write address	
AWLEN_M[7:0]	Output	Write burst length	
AWSIZE_M[2:0]	Output	Write burst size	
AWBURST_M[1:0]	Output	Write burst type	
AWLOCK_M	Output	Write lock type	
AWCACHE_M[3:0]	Output	Write cache type	
AWUSER_M[n:0][15]	Output	User signal	
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	-	
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 slave device, if populated.

[13] For HN-I, x = 10. For HN-P, x = 19. For SBSX, x = 23. For more information, see B.4.4 Calculating the SBSX AxID signal widths on page 973.

[14] The value of n is configuration-dependent.

[15] For HN-I, HN-D, HN-P, and SBSX, n = *`REQ_RSVDC_WIDTH`*.

Signal	Direction	Description	Connection information
AWIDUNQ_M	Output	Unique ID indicator signal	Connect to corresponding slave device, if populated.
AWNSAID_M[3:0]	Output	Non-secure Access Identifier signal	Connect to corresponding slave device, if populated.
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 slave device, if populated.
WREADY_M	Input	Write data ready	Connect to corresponding slave device, if populated, otherwise tie LOW.
WVALID_M	Output	Write data valid	Connect to corresponding slave device, if populated.
WDATA_M[n:0][14]	Output	Write data	Connect to corresponding slave device, if populated.
WSTRB_M[n:0][14]	Output	Write byte lane strobes	Connect to corresponding slave device, if populated.
WLAST_M	Output	Write data last transfer indication	Connect to corresponding slave device, if populated.
WUSER_M[u:0][16]	Output	**WUSER_M[0]** is **WDATACHK** valid signal	Connect to corresponding slave device, if populated.
WPOISON_M[p:0] [17]	Output	Poison signal	Connect to corresponding master device, if populated, otherwise tie LOW.
WDATACHK_M[d:0] [18]	Output	Data check signal	Connect to corresponding master device, if populated, otherwise tie LOW.
WTRACE_M	Output	Trace signal	Connect to corresponding master device, if populated, otherwise tie LOW.
BREADY_M	Output	Write response ready	Connect to corresponding slave device, if populated.
BVALID_M	Input	Write response valid	Connect to corresponding slave device, if populated, otherwise tie LOW.
BID_M[x:0][13]	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 slave device, if populated, otherwise tie LOW.

[16] 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.

[17] The value of *p* = (((n + 1) / 64) - 1).

[18] The value of *d* = (((n + 1) / 8) - 1).

Signal	Direction	Description	Connection information
BCOMP_M	Input	Write/CMO observable. This signal is only present on SBSX. It is not present on HN-I.	Connect to corresponding slave 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 slave device, if populated, otherwise tie LOW.
ARREADY_M	Input	Read address ready	Connect to corresponding slave device, if populated, otherwise tie LOW.
ARVALID_M	Output	Read address valid	Connect to corresponding slave device, if populated.
ARID_M[x:0][13]	Output	Read address ID	
ARADDR_M[n:0][14]	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, where n = (REQ_RSVDC_WIDTH-1).	User signal	
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 slave device, if populated.
ARIDUNQ_M	Output	Unique ID indicator signal	Connect to corresponding slave device, if populated.
ARNSAID_M[3:0]	Output	Non-secure Access Identifier signal	Connect to corresponding slave device, if populated.
RREADY_M	Output	Read data ready	Connect to corresponding slave device, if populated.
RVALID_M	Input	Read data valid	Connect to corresponding slave device, if populated, otherwise tie LOW.

Signal	Direction	Description	Connection information
RID_M[x:0][13]	Input	Read data ID	Connect to corresponding slave device, if populated, otherwise tie LOW.
RDATA_M[127:0]/[255:0]	Input	Read data	Connect to corresponding slave device, if populated, otherwise tie LOW.
RRESP_M[1:0]	Input	Read data response	Connect to corresponding slave device, if populated, otherwise tie LOW.
RLAST_M	Input	Read data last transfer indication	Connect to corresponding slave device, if populated, otherwise tie LOW.
RUSER_M[u:0][16]	Input	**RUSER_M[0]** is **RDATACHK** valid signal. If *META_DATA_EN* = 1, tie **RUSER_M[u:1]** to 0.	Connect to corresponding slave device, if populated, otherwise tie LOW.
RPOISON_M[p:0] [17]	Input	Poison signal	Connect to corresponding master device, if populated, otherwise tie LOW.
RDATACHK_M[d:0][18]	Input	Data check signal	Connect to corresponding master device, if populated.
RTRACE_M	Input	Trace signal	Connect to corresponding master device, if populated, otherwise tie LOW.
RIDUNQ_M	Input	Unique ID indicator signal	Connect to corresponding master 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.4.4 Calculating the SBSX AxID signal widths

By default, the width of the **AxID** signals in SBSX is 24 bits. However, you can modify specific CMN-650 properties to reduce the number of bits used for **AxID** signals for tracker optimization purposes.

Use the following equations to calculate the **AWID** width.

Equations when CMO_ON_WRITE is enabled

If SBSX *CMO_ON_AW* = 1:

- **AWID** = (2part_pcmo_coloring + PGroupID + $\log_2$(NUM(RNF+RNI+RND+CXHA+HNF)) + $\log_2$(ReqTracker_size))
- For example, the size of **AWID** based on the maximum values for each of the preceding parameters = 1 + 1 + 1 + 11 + 5 + 5 = 24.

AWID encoding: {Reserved[0:0], Reserved[0:0], 2-part_PCMO[0:0], ReturnNID[log2(NUM(RNF+RNI+RND+CXHA+HNF))-1:0], PGroupID[4:0], RequestTrackerID[log2(*SBSX_NUM_REQS*)-1:0]}

ARID encoding: {DartID[log2(*SBSX_NUM_DART*)-1:0]}

Note

2part_pcmo_coloring indicates if the PCMO is a single part or two part PCMO on CHI.

Equations when CMO_ON_WRITE is disabled

If CMO_ON_WRITE is disabled:

- **AWID** = $\log_2$(ReqTracker_size)
- For example, the size of **AWID** based on the maximum value for the preceding parameter = 5.

The following equation indicates the maximum number of usable bits for **ARID**:

- **ARID** (usable bits) = $\log_2$(NUM_DART)

For CMN-650, MTU color is **ARID**[MSB].

B.4.5 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 either 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.

The following table shows the **AxID** properties and encodings for the HN-I and HN-P.

Table B-8: 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	20	physical_mem_en = 0	Non peer-to-peer	AWID or ARID	**AxID[10:0]** Same as HN-I Device memory **AxID[19:11]** Reserved
		physical_mem_en = 1	Non peer-to-peer	AWID or ARID	**AxID[10:0]** Same as HN-I Normal memory **AxID[19:11]** Reserved
		All	Peer-to-peer	AWID	**AWID[3:0]** Hash of PCIe RN-I AWID **AWID[5:4]** RN-I port number **AWID[11:6]** RN-I Logical ID **AWID[15:12]** CXHA Logical ID **AWID[16]** Remote RN-I **AWID[18:17]** Reserved **AWID[19]** Peer-to-peer coloring

Device type	AxID width	Downstream memory type	Request source	AW or AR	Encoding
		All	Peer-to-peer	ARID	ARID[***HNP_RD_NUM_AXI_RE\QS_PARAM_LOG2***-1:0] HN-P read RDT entry ARID[18:***HNP_RD_NUM_AXI_RE\QS_PARAM_LOG2***] Reserved AWID[19] Peer-to-peer coloring

B.4.6 A4S signals

The A4S interface signals are listed in the following tables.

Signal definitions

Table B-9: A4S Transmit and Receive signals

Signal	Direction	Description	Connection information
TXA4STREADY	Input	**TXA4STREADY** indicates that the slave can accept a transfer in the current cycle.	Connect from **RXA4STREADY** of the A4S slave, if populated, otherwise tie LOW.
TXA4STVALID	Output	**TXA4STVALID** indicates that the master is driving a valid transfer. A transfer takes place when both **TXA4STVALID** and **TXA4STREADY** are asserted.	Connect to **RXA4STVALID** of the A4S slave, if populated.
TXA4STDEST[7:0]	Output	`0b00000000`	**TXA4STDEST** is always zero.
TXA4STID[7:0]	Output	**TXA4STID** is the data stream identifier that indicates different streams of data.	Connect to **RXA4STID** of the A4S slave, 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 slave, if populated.
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 slave, 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 slave, if populated.
TXA4STLAST	Output	**TXA4STLAST** indicates the boundary of a packet.	Connect to **RXA4STLAST** of the A4S slave, if populated.
RXA4STREADY	Output	**RXA4STREADY** indicates that the slave can accept a transfer in the current cycle.	Connect to **TXA4STREADY** of the A4S master, if populated.

Signal	Direction	Description	Connection information
RXA4STVALID	Input	**RXA4STVALID** indicates that the master is driving a valid transfer. A transfer takes place when both **RXA4STVALID** and **RXA4STREADY** are asserted.	Connect from **TXA4STVALID** of the A4S master, if populated, otherwise tie LOW.
RXA4STDEST[7:0]	Input	**RXA4STDEST** provides routing information for the data stream.	Connect from **TXA4STDEST** of the A4S master, if populated, otherwise tie LOW.
RXA4STID[7:0]	Input	**RXA4STID** is the data stream identifier that indicates different streams of data.	Connect from **TXA4STID** of the A4S master, if populated, otherwise tie LOW.
RXA4STRI[7:0]	Input	**RXA4STRI** is the chip to chip routing information that indicates RA ID of the other chip.	Connect from **TXA4STRI** of the A4S master, 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 master, 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 master, 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 master, if populated, otherwise tie LOW.
RXA4STLAST	Input	**RXA4STLAST** indicates the boundary of a packet.	Connect from **TXA4STLAST** of the A4S master, if populated, otherwise tie LOW.

B.5 APB interface signals

HN-D nodes have an APB interface to support the connection of an external APB master device.

Signal definitions

Table B-10: 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 master device.
PPROT[2:0]	Input	Protection type of the transaction.	
PSEL	Input	Indicates that the slave 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.	
PSTRB[3:0]	Input	Write strobes.	
PREADY	Output	Ready.	
PRDATA[31:0]	Output	Read data.	
PSLVERR	Output	Indicates a transfer failure.	

B.6 RN-D ACE-Lite-with-DVM device population signals

Device population signals are present only if you configure CMN-650 to include the relevant RN-D bridge.

Signal definitions

Table B-11: RN-D ACE-Lite-with-DVM device population signals

Signal	Direction	Description	Connection information
ACCHANNELEN_S0_NID<x>	Input	Indicates that: • The RN-D bridge at NodeID `<x>`is populated. • AMBA® slave port 0 for NodeID `<x>` is of type ACE-Lite+DVM and includes a device which responds to DVM messages on the AC channel. **0** DVM-capable device is not populated. **1** DVM-capable device is populated.	Tie as required for system configuration.

Signal	Direction	Description	Connection information
ACCHANNELEN_S1_NID<x>	Input	Indicates that: • The RN-D bridge at NodeID `<x>` is populated. • AMBA® slave port 1 for NodeID `<x>` is of type ACE-Lite+DVM and includes a device which responds to DVM messages on the AC channel. **0** DVM-capable device is not populated. **1** DVM-capable device is populated.	
ACCHANNELEN_S2_NID<x>	Input	Indicates that: • The RN-I bridge at NodeID `<x>` is populated. • AMBA® slave port 2 for NodeID `<x>` is of type ACE-Lite+DVM and includes a device which responds to DVM messages on the AC channel. **0** DVM-capable device is not populated. **1** DVM-capable device is populated.	

B.7 Debug, trace, and PMU interface signals

Signals that aid debugging are included in CMN-650.

Note

All signal names in this section are only a root name indicated as **RootName**. CMN-650 interfaces use **RootName** within a more fully specified signal name as follows:

CMN-650 interface signal name == **RootName_NID#**, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-12: 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.	-

Signal	Direction	Description	Connection information
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.

Signal	Direction	Description	Connection information
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.
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.8 ATPG interface signals

Signals that support ATPG capabilities are included in CMN-650.

Signal definitions

Table B-13: DFT signals

Signal	Direction	Description	Connection information
DFTCLKBYPASS	Input	Select the SLC RAM clock to follow the CMN-650 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.	

Signal	Direction	Description	Connection information
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. They are held the cycle that the RAM clock asserts. The RAM hold constraints are not guaranteed during ATPG test. The constraints are not guaranteed because random data is shifted into the flops that control the set address and write data flop enables. Therefore, 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.8.1 Block-level ATPG signals

CMN-650 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 por_hnf block-level.

Table B-14: HN-F block-level por_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-15: 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`.

Signal	Direction	Description
DFTCGEN	Input	Overrides clock gate shift. 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-16: 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`.
DFTMCPHOLD	Input	Limits number of multicycle path toggles during ATPG delay tests. Functional mode = `0b0`.
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-17: 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-18: 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-19: 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-20: 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-21: 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 CXRH block-level named por_cxrh.

Table B-22: CXRH block-level por_cxrh 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`.
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`.
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 at the CXLA block-level named por_cxla.

Table B-23: CXLA block-level por_cxla 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`.
DFTMCPHOLD	Input	Limits number of multicycle path toggles during ATPG delay tests. Functional mode = `0b0`.
CLK_CGL, CLK_CXS	Input	Functional clocks.
nRESET_CGL, nRESET_CXS	Input	Functional resets. 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-24: 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.

B.9 MBIST interface signals

Signals that support MBIST capabilities are included in CMN-650.

Signal definitions

Table B-25: 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.10 Clock management signals

This section contains information on clock management Q-Channel signals in the CMN-650.

Signal definitions

Table B-26: Clock management Q-Channel signals

Signal	Direction	Description	Connection information
QACTIVE_CLKCTL	Output	Indication that CMN-650 is active, and that the *External Clock Controller* (ExtCC) must not make a request for CMN-650 to prepare to stop the clocks.	Connect to external clock controller.

Signal	Direction	Description	Connection information
QREQn_CLKCTL	Input	Request from the ExtCC for the CMN-650 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-650 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-650 has refused the request from the ExtCC to prepare to stop the clocks.	

B.10.1 CML clock management signals

This section contains information on clock management Q-Channel signals when using CML.

Signal definitions

Table B-27: Clock management Q-Channel signals

Signal	Direction	Description	Connection information
QACTIVE_CGLCLKCTL	Output	Indication that the CGL side of CMN-650 is active, and that the *External Clock Controller* (ExtCC) must not make a request for CMN-650 and corresponding CXG device to prepare to stop the clock **CLK_CGL**.	OR with **QACTIVE_CGLCLKCTL** (CXLA) and connect to external clock controller.
QREQn_CGLCLKCTL	Input	Request from the ExtCC for the CMN-650 to prepare to stop the clock **CLK_CGL**. Active-LOW.	Connect to external clock controller or tie HIGH if unused.
QACCEPTn_CGLCLKCTL	Output	Positive acknowledgment after receiving **QREQn** assertion indicating that CMN-650 has completed preparation to stop the clock **CLK_CGL** and that the ExtCC can stop the clock **CLK_CGL**. Active-LOW.	Connect to external clock controller.
QDENY_CGLCLKCTL	Output	Negative acknowledgment after receiving **QREQn** assertion indicating that CMN-650 has refused the request from the ExtCC to prepare to stop the clock **CLK_CGL**.	

B.11 Power management signals

This section contains information on power management signals for the logic power domain.

Signal definitions

Table B-28: 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.

Signal	Direction	Description	Connection information
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-650.	Connect to external power management controller.
PDENY_LOGIC	Output	Indicates denial of the power state transition.	
PACTIVE_LOGIC	Output	Hint that indicates activity across the CMN-650. When LOW, indicates the possibility of entering static retention or the OFF state.	

If **PACTIVE_LOGIC** is asserted, the system cannot be powered down.

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-650 interfaces use **RootName** within a fully specified signal name as follows:

CMN-650 interface signal name == **RootName_NID#**, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-29: 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.	
INTREQFAULTNS	Output	Non-secure fault handling interrupt.	
INTREQFAULTS	Output	Secure fault handling interrupt.	
INTREQMPAMERRNS	Output	Non-secure *Memory System Resource Partitioning and Monitoring* (MPAM) fault handling interrupt.	
INTREQMPAMERRS	Output	Secure MPAM fault handling interrupt.	

[19] If MCP, the MCP duration must be ≤8 cycles to the last flop to receive this signal. This constraint is a requirement for implementation.

B.13 Configuration input signals

CMN-650 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-650, 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-650 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.

B.14 Processor event interface signals

Signals that support processor event interface capabilities are included in CMN-650. Processor event interface signals are present at RN-F, RN-I, and RN-D node locations.

Note

All signal names in this section are only a root name indicated as **RootName**. CMN-650 interfaces use **RootName** within a more fully specified signal name as follows:

CMN-650 interface signal name == **RootName_NID#**, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-31: 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-650 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-650 must handle **EVENT_OUT** from CHI processor. **EVENT_OUT** is a multicycle pulse. If system integration wants to connect the **EVENT_OUT** to CMN-650 **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-650 can drive **EVENT_IN** of CHI processor.

B.15 CCIX Gateway Link interface signals

CMN-650 includes *CCIX Gateway Link* (CGL) interface signals. The CGL interface is a credited micro-architecture interface between CXRA, CXHA, and CXLA components.

Note

All signal names in this section are only a root name indicated as **RootName**. CMN-650 interfaces use **RootName** within a more fully specified signal name as follows:

CMN-650 interface signal name == **RootName_NID#**, where # represents the node ID corresponding to the specific interface.

Signal definitions

Table B-32: CGL interface signals

Signal	Direction	Description	Connection information
TXCGLREQDATFLITPEND	Output	Transmit Memory Request early flit valid hint	Connect to **RXCGLREQDATFLITPEND** of the corresponding CXLA.
TXCGLREQDATFLITV	Output	Transmit Memory Request flit valid	Connect to **RXCGLREQDATFLITV** of the corresponding CXLA.
TXCGLREQDATFLIT[n:0][20]	Output	Transmit Memory Request flit	Connect to **RXCGLREQDATFLIT** of the corresponding CXLA.
TXCGLREQDATLCRDV	Input	Transmit Memory Request channel link layer credit	Connect to **RXCGLREQDATLCRDV** of the corresponding CXLA.
TXCGLSNPFLITPEND	Output	Transmit Snoop Request early flit valid hint	Connect to **RXCGLSNPFLITPEND** of the corresponding CXLA.
TXCGLSNPFLITV	Output	Transmit Snoop Request flit valid	Connect to **RXCGLSNPFLITV** of the corresponding CXLA.

[20] The value n is configuration-dependent.

Signal	Direction	Description	Connection information
TXCGLSNPFLIT[m:0][21]	Output	Transmit Snoop Request flit	Connect to **RXCGLSNPFLIT** of the corresponding CXLA.
TXCGLSNPLCRDV	Input	Transmit Snoop Request channel link layer credit	Connect to **RXCGLSNPLCRDV** of the corresponding CXLA.
TXCGLREQRSPFLITPEND	Output	Transmit Memory Response early flit valid hint	Connect to **RXCGLREQRSPFLITPEND** of the corresponding CXLA.
TXCGLREQRSPFLITV	Output	Transmit Memory Response flit valid	Connect to **RXCGLREQRSPFLITV** of the corresponding CXLA.
TXCGLREQRSPFLIT[35:0]	Output	Transmit Memory Response flit	Connect to **RXCGLREQRSPFLIT** of the corresponding CXLA.
TXCGLREQRSPLCRDV	Input	Transmit Memory Response channel link layer credit	Connect to **RXCGLREQRSPLCRDV** of the corresponding CXLA.
TXCGLSNPRSPFLITPEND	Output	Transmit Snoop Response early flit valid hint	Connect to **RXCGLSNPRSPFLITPEND** of the corresponding CXLA.
TXCGLSNPRSPFLITV	Output	Transmit Snoop Response flit valid	Connect to **RXCGLSNPRSPFLITV** of the corresponding CXLA.
TXCGLSNPRSPFLIT[35:0]	Output	Transmit Snoop Response flit	Connect to **RXCGLSNPRSPFLIT** of the corresponding CXLA.
TXCGLSNPRSPLCRDV	Input	Transmit Snoop Response channel link layer credit	Connect to **RXCGLSNPRSPLCRDV** of the corresponding CXLA.
TXCGLSNPDATFLITPEND	Output	Transmit Snoop Data early flit valid hint	Connect to **RXCGLSNPDATFLITPEND** of the corresponding CXLA.
TXCGLSNPDATFLITV	Output	Transmit Snoop Data flit valid	Connect to **RXCGLSNPDATFLITV** of the corresponding CXLA.
TXCGLSNPDATFLIT[559:0]	Output	Transmit Snoop Data flit	Connect to **RXCGLSNPDATFLIT** of the corresponding CXLA.
TXCGLSNPDATLCRDV	Input	Transmit Snoop Data channel link layer credit	Connect to **RXCGLSNPDATLCRDV** of the corresponding CXLA.
TXCGLRSPDATFLITPEND	Output	Transmit Memory Response Data early flit valid hint	Connect to **RXCGLRSPDATFLITPEND** of the corresponding CXLA.

[21] The value m is configuration-dependent.

Signal	Direction	Description	Connection information
TXCGLRSPDATFLITV	Output	Transmit Memory Response Data flit valid	Connect to **RXCGLRSPDATFLITV** of the corresponding CXLA.
TXCGLRSPDATFLIT[559:0]	Output	Transmit Memory Response Data flit	Connect to **RXCGLRSPDATFLIT** of the corresponding CXLA.
TXCGLRSPDATLCRDV	Input	Transmit Memory Response Data channel link layer credit	Connect to **RXCGLRSPDATLCRDV** of the corresponding CXLA.
TXCGLPCRDFLITPEND	Output	Transmit Protocol Credit early flit valid hint	Connect to **RXCGLPCRDFLITPEND** of the corresponding CXLA.
TXCGLPCRDFLITV	Output	Transmit Protocol Credit flit valid	Connect to **RXCGLPCRDFLITV** of the corresponding CXLA.
TXCGLPCRDFLIT[50:0]	Output	Transmit Protocol Credit flit	Connect to **RXCGLPCRDFLIT** of the corresponding CXLA.
TXCGLPCRDLCRDV	Input	Transmit Protocol Credit channel link layer credit	Connect to **RXCGLPCRDLCRDV** of the corresponding CXLA.
RXCGLREQDATFLITPEND	Input	Receive Memory Request early flit valid hint	Connect to **TXCGLREQDATFLITPEND** of the corresponding CXLA.
RXCGLREQDATFLITV	Input	Receive Memory Request flit valid	Connect to **TXCGLREQDATFLITV** of the corresponding CXLA.
RXCGLREQDATFLIT[n:0][20]	Input	Receive Memory Request flit	Connect to **TXCGLREQDATFLIT** of the corresponding CXLA.
RXCGLREQDATLCRDV	Output	Receive Memory Request channel link layer credit	Connect to **TXCGLREQDATLCRDV** of the corresponding CXLA.
RXCGLSNPFLITPEND	Input	Receive Snoop Request early flit valid hint	Connect to **TXCGLSNPFLITPEND** of the corresponding CXLA.
RXCGLSNPFLITV	Input	Receive Snoop Request flit valid	Connect to **TXCGLSNPFLITV** of the corresponding CXLA.
RXCGLSNPFLIT[m:0][21]	Input	Receive Snoop Request flit	Connect to **TXCGLSNPFLIT** of the corresponding CXLA.
RXCGLSNPLCRDV	Output	Receive Snoop Request channel link layer credit	Connect to **TXCGLSNPLCRDV** of the corresponding CXLA.
RXCGLREQRSPFLITPEND	Input	Receive Memory Response early flit valid hint	Connect to **TXCGLREQRSPFLITPEND** of the corresponding CXLA.

Signal	Direction	Description	Connection information
RXCGLREQRSPFLITV	Input	Receive Memory Response flit valid	Connect to **TXCGLREQRSPFLITV** of the corresponding CXLA.
RXCGLREQRSPFLIT[35:0]	Input	Receive Memory Response flit	Connect to **TXCGLREQRSPFLIT** of the corresponding CXLA.
RXCGLREQRSPLCRDV	Output	Receive Memory Response channel link layer credit	Connect to **TXCGLREQRSPLCRDV** of the corresponding CXLA.
RXCGLSNPRSPFLITPEND	Input	Receive Snoop Response early flit valid hint	Connect to **TXCGLSNPRSPFLITPEND** of the corresponding CXLA.
RXCGLSNPRSPFLITV	Input	Receive Snoop Response flit valid	Connect to **TXCGLSNPRSPFLITV** of the corresponding CXLA.
RXCGLSNPRSPFLIT[35:0]	Input	Receive Snoop Response flit	Connect to **TXCGLSNPRSPFLIT** of the corresponding CXLA.
RXCGLSNPRSPLCRDV	Output	Receive Snoop Response channel link layer credit	Connect to **TXCGLSNPRSPLCRDV** of the corresponding CXLA.
RXCGLSNPDATFLITPEND	Input	Receive Snoop Data early flit valid hint	Connect to **TXCGLSNPDATFLITPEND** of the corresponding CXLA.
RXCGLSNPDATFLITV	Input	Receive Snoop Data flit valid	Connect to **TXCGLSNPDATFLITV** of the corresponding CXLA.
RXCGLSNPDATFLIT[559:0]	Input	Receive Snoop Data flit	Connect to **TXCGLSNPDATFLIT** of the corresponding CXLA.
RXCGLSNPDATLCRDV	Output	Receive Snoop Data channel link layer credit	Connect to **TXCGLSNPDATLCRDV** of the corresponding CXLA.
RXCGLRSPDATFLITPEND	Input	Receive Memory Response Data early flit valid hint	Connect to **TXCGLRSPDATFLITPEND** of the corresponding CXLA.
RXCGLRSPDATFLITV	Input	Receive Memory Response Data flit valid	Connect to **TXCGLRSPDATFLITV** of the corresponding CXLA.
RXCGLRSPDATFLIT[559:0]	Input	Receive Memory Response Data flit	Connect to **TXCGLRSPDATFLIT** of the corresponding CXLA.
RXCGLRSPDATLCRDV	Output	Receive Memory Response Data channel link layer credit	Connect to **TXCGLRSPDATLCRDV** of the corresponding CXLA.
RXCGLPCRDFLITPEND	Input	Receive Protocol Credit early flit valid hint	Connect to **TXCGLPCRDFLITPEND** of the corresponding CXLA.

Signal	Direction	Description	Connection information
RXCGLPCRDFLITV	Input	Receive Protocol Credit flit valid	Connect to **TXCGLPCRDFLITV** of the corresponding CXLA.
RXCGLPCRDFLIT[50:0]	Input	Receive Protocol Credit flit	Connect to **TXCGLPCRDFLIT** of the corresponding CXLA.
RXCGLPCRDLCRDV	Output	Receive Protocol Credit channel link layer credit	Connect to **TXCGLPCRDLCRDV** of the corresponding CXLA.
RXCGLLINKACTIVEREQ	Input	Receive channel LinkActive request from CXLA	Connect to **TXCGLLINKACTIVEREQ** of the corresponding CXLA.
RXCGLLINKACTIVEACK	Output	Receive channel LinkActive acknowledgment to CXLA	Connect to **TXCGLLINKACTIVEACK** of the corresponding CXLA.
TXCGLLINKACTIVEREQ	Output	Transmit channel LinkActive request to CXLA	Connect to **RXCGLLINKACTIVEREQ** of the corresponding CXLA.
TXCGLLINKACTIVEACK	Input	Transmit channel LinkActive acknowledgment from CXLA	Connect to **RXCGLLINKACTIVEACK** of the corresponding CXLA.
RXCGLSACTIVE	Input	Indication from CXLA that it has one or more outstanding protocol-layer transactions. **RXSACTIVE** must remain asserted throughout the lifetime of the transaction.	Connect to **TXCGLSACTIVE** of the corresponding CXLA.
TXCGLSACTIVE	Output	Indication to CXLA that CXRH has one or more outstanding protocol-layer transactions. **TXCGLSACTIVE** remains asserted throughout the lifetime of the transaction.	Connect to **RXCGLSACTIVE** of the corresponding CXLA.

B.16 CXLA configuration interface signals

CMN-650 includes signals to support CXLA configuration.

Note

All signal names in this section are only a root name indicated as **RootName**. CMN-650 interfaces use **RootName** within a more fully specified signal name as follows:

CMN-650 interface signal name == **RootName_NID#**, where # represents the node ID corresponding to the specific interface.

Note

For the top-level CXLA interface signals, see the *Configuration and Integration Manual*, which is only available to licensees.

Signal definitions

Table B-33: CXLA configuration interface signals

Signal	Direction	Description	Connection information
RXPUBFLITPEND	Input	Receive channel early flit valid hint	Connect to **TXPUBFLITPEND** of the corresponding CXLA device, if populated. Otherwise, tie LOW.
RXPUBFLITV	Input	Receive channel flit valid	Connect to **TXPUBFLITV** of the corresponding CXLA device, if populated. Otherwise, tie LOW.
RXPUBFLIT[34:0]	Input	Receive channel flit	Connect to **TXPUBFLIT[n:0]** of the corresponding CXLA device, if populated. Otherwise, tie LOW.
RXPUBLINKFLIT	Input	Receive channel link flit	Connect to **TXPUBLINKFLIT** of the corresponding CXLA device, if populated. Otherwise, tie LOW.
RXPUBLCRDV_RP1	Output	Receive channel link layer credit	Connect to **TXPUBLCRDV_RP1** of the corresponding CXLA device, if populated.
RXPUBLINKACTIVEREQ	Input	Receive channel LinkActive request	Connect to **TXPUBLINKACTIVEREQ** of the corresponding CXLA device, if populated. Otherwise, tie LOW.
RXPUBLINKACTIVEACK	Output	Receive channel LinkActive acknowledge	Connect to **TXPUBLINKACTIVEACK** of the corresponding CXLA device, if populated.
TXPUBFLITPEND	Output	Transmit channel flit valid	Connect to **RXPUBFLITPEND** of the corresponding CXLA device, if populated.
TXPUBFLITV	Output	Transmit channel early flit valid hint	Connect to **RXPUBFLITV** of the corresponding CXLA device, if populated.
TXPUBFLIT[34:0]	Output	Transmit channel flit	Connect to **RXPUBFLIT[34:0]** of the corresponding CXLA device, if populated.
TXPUBLINKFLIT	Output	Transmit channel link flit	Connect to **RXPUBLINKFLIT** of the corresponding CXLA device, if populated.
TXPUBLCRDV_RP1	Input	Transmit channel link layer credit	Connect to **RXPUBLCRDV_RP1** of the corresponding CXLA device, if populated, otherwise tie LOW.
TXPUBLINKACTIVEREQ	Output	Transmit channel LinkActive request	Connect to **RXPUBLINKACTIVEREQ** of the corresponding CXLA device, if populated.
TXPUBLINKACTIVEACK	Input	Transmit channel LinkActive acknowledge	Connect to **RXPUBLINKACTIVEACK** of the corresponding CXLA device, if populated, otherwise tie LOW.
TXPUBCFGACTIVE	Output	Transmit channel configuration active	Connect to **RXPUBCFGACTIVE** of the corresponding CXLA device, if populated.

Appendix C Revisions

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

Table C-1: Issue 0000-00

Change	Location
First release	-

Table C-2: Differences between issue 0000-00 and issue 0100-01

Change	Location
Updated feature list.	2.3 Features on page 21
Added Protocol feature compliance appendix and moved description of CML CXS, CCIX, and CHI feature support to that appendix.	A Protocol feature compliance on page 950
Added description of AXI and CHI feature support.	• A.1 AXI and ACE-Lite feature support on page 950 • A.2 CHI feature support on page 951
Added description of new CHI-C memory interface support.	3.6 System component selection on page 51
Updated mesh sizing and top-level configuration options.	3.7 Deciding on the size of the mesh on page 54
Updated device placement and configuration options.	Change superseded by later version.
Updated product revision history with details of r1p0 changes.	2.8 Product revisions on page 31
Clarified definition of MXP XID and YID.	3.2 Crosspoint (XP) on page 34
Added description of APB interface support.	Some changes superseded by later version. • 5.1 About the programmers model on page 223 • 5.1.1 Node configuration register address mapping on page 223 • B.5 APB interface signals on page 977
Added description of support for multiple asynchronous clock domains and the impact on global clocks.	Some changes superseded by later version. • 4.1.1 Clock domain configurations on page 63 • 4.1.3 Clock hierarchy on page 67 • 4.1.4 Clock enable inputs on page 68
Updated CAL description and added HN-I and SBSX to supported device types.	3.4.6 Component Aggregation Layer (CAL) on page 46
Moved example system diagram containing credited slices.	3.4.7 Credited Slices (CSs) on page 47
Moved description of credited slice types.	• 3.4.7.1 Mesh Credited Slice (MCS) on page 48 • 3.4.7.3 Device Credited Slice (DCS) on page 50 • 3.4.7.4 CAL Credited Slice (CCS) on page 50
Added description of *Asynchronous Mesh Credited Slice* (AMCS) component.	3.4.7.2 Asynchronous Mesh Credited Slice (AMCS) on page 48
Added *CHI Domain Bridge* (CDB) and *AMBA Domain Bridge* (ADB) topics.	Change superseded by future version
Added description of dual DAT/RSP feature.	4.10.1 Dual DAT and RSP channels on page 139
Updated description of ROOTNODEBASE structure and examples for clarity.	4.12.1 Configuration address space organization on page 169

Change	Location
Updated description of configuration register node structure for clarity.	4.12.2 Configuration register node structure on page 171
Corrected description of child pointers to clarify which devices use the external child node type.	4.12.3 Child pointers on page 174
Updated description of exclusive accesses to clarify opcodes that are used for shareable and coherent memory locations.	• 4.18.2.1 Exclusive accesses in HN-F on page 204 • 6.4.2 Cacheable and non-cacheable exclusives on page 889
Added a new constraint to description of CML exclusive support.	4.18.2.3 CML support for exclusive accesses on page 204
Updated description of processor events to clarify that event interface signals are present on RN-I and RN-D nodes.	4.19 Processor events on page 213
Updated PCIe integration section and clarified the difference between PCIe HN-Is and other HN-Is.	4.15 PCIe integration on page 181
Corrected error in error interrupt handler flow example figure.	Figure 4-53: Error interrupt handler flow example on page 187
Updated description of Completer Busy indication to provide example tracker occupancy levels and description of RA CBusy handling.	4.18.5 Completer Busy indication on page 207
Added description of advanced CBusy handling in HN-F.	4.18.5.1 Advanced CBusy handling in HN-F on page 208
Corrected maximum number of physical address bits in various places.	• 4.5.4 System Cache Groups (SCGs) on page 104 • 4.5.2 SAM memory region size configuration on page 100
Corrected maximum number of HN-Fs and CAL instances in normal mode.	4.5.7 Support for HN-Fs with CAL in the RN SAM on page 110
Added section describing QoS override based on memory regions.	4.5.10 Address range-based QoS override and PrefetchTgt support on page 113
Updated description of HN-I SAM and example HN-I SAM configuration for clarity.	• 4.9 HN-I SAM on page 129 • 4.9.3 HN-I SAM example configuration on page 131
Added description of default mesh XY routing algorithm.	4.10.5 Default XY routing behavior on page 145
Added description of non-XY routing feature.	4.10.6 Non-XY routing on page 146
Updated various sections to describe configuring the CXS logic to be synchronous or asynchronous to the global clock.	• 4.1.2 CML clock inputs on page 66 • 4.1.7 CML clock management on page 71
Added diagram of an example CML system power domain configuration without CXDB.	Figure 4-9: Single CML power domain example on page 76
Added description of configuring CML power domains using *DB_PRESENT*.	• 4.2.2 Power domain control on page 76 • 4.2.4 CXS power domain on page 79
Corrected a typo in the described system coherency states in the hardware interface description.	4.2.9.1 Hardware interface on page 86
Updated description of configuration register accesses to indicate support for AXI and APB.	5.1 About the programmers model on page 223
Updated performance monitoring event encodings for the pmu_event_sel registers of various blocks.	• 5.3.1.14 por_rnd_pmu_event_sel on page 260 • 5.3.4.39 por_cxla_pmu_event_sel on page 372 • 5.3.13.14 por_rni_pmu_event_sel on page 647 • 5.3.14.17 por_mxp_pmu_event_sel on page 668 • 5.3.15.94 por_hnf_pmu_event_sel on page 816
Updated various HN-I register descriptions.	• 5.3.3.9 por_hni_cfg_ctl on page 305 • 5.3.3.4 por_hni_unit_info on page 298

Change	Location
Added CXLA register descriptions.	• 5.3.4.5 por_cxla_cfg_ctl on page 328 • 5.3.4.21 por_cxla_tlp_hdr_fields on page 352
Updated various CXLA register descriptions.	• 5.3.4.3 por_cxla_secure_register_groups_override on page 326 • 5.3.4.6 por_cxla_aux_ctl on page 329 • 5.3.4.8 por_cxla_ccix_prop_configured on page 335 • 5.3.4.20 por_cxla_linkid_to_pcie_bus_num on page 351
Added a DN register description.	5.3.5.5 por_dn_cfg_ctl on page 382
Updated various DN register descriptions.	• 5.3.5.3 por_dn_build_info on page 380 • 5.3.5.4 por_dn_secure_register_groups_override on page 381 • 5.3.5.6 por_dn_aux_ctl on page 383
Added various RN SAM register descriptions.	5.3.7.27 sam_qos_mem_region_reg0-7 on page 476
Updated an RN SAM register description.	5.3.7.4 por_rnsam_unit_info on page 441
Corrected region{x}_size field reset value to a 7-bit value for the following registers: • sys_cache_grp_region{0-3} • sys_cache_grp_secondary_reg{0-3} • non_hash_mem_region_reg{0-19}	5.3.7 RN SAM register descriptions on page 438
Updated an HN-F MPAM_S register description.	5.3.8.4 por_hnf_mpam_sidr on page 482
Updated various configuration master register descriptions.	• 5.3.9.29 por_info_global on page 532 • 5.3.9.34 por_mpam_ns_err_int_status on page 537
Added various SBSX register descriptions.	• 5.3.11.3 por_sbsx_secure_register_groups_override on page 573 • 5.3.11.5 por_sbsx_cfg_ctl on page 576
Updated an SBSX register description.	5.3.11.4 por_sbsx_unit_info on page 574
Updated an HN-F MPAM_NS register description.	5.3.12.16 por_hnf_ns_mpam_esr on page 608
Added various MXP register descriptions.	• 5.3.14.42 por_mxp_multi_dat_rsp_chn_sel_0-15 on page 702 • 5.3.14.43 por_mxp_multi_dat_rsp_chn_ctrl on page 704 • 5.3.14.44 por_mxp_xy_override_sel_0-7 on page 705
Updated various MXP register descriptions.	• 5.3.14.2 por_mxp_device_port_connect_info_p0-1 on page 650 • 5.3.14.13 por_mxp_p0-1_qos_control on page 664 • 5.3.14.14 por_mxp_p0-1_qos_lat_tgt on page 665 • 5.3.14.15 por_mxp_p0-1_qos_lat_scale on page 666 • 5.3.14.16 por_mxp_p0-1_qos_lat_range on page 667 • 5.3.14.17 por_mxp_pmu_event_sel on page 668 • 5.3.14.21 por_mxp_errmisc on page 674 • 5.3.14.26 por_mxp_errmisc_NS on page 680 • 5.3.14.27 por_mxp_p0-1_syscoreq_ctl on page 681 • 5.3.14.28 por_mxp_p0-1_syscoack_status on page 683

Change	Location
Added various HN-F register descriptions.	• 5.3.15.68 por_hnf_sam_sn_properties1 on page 788 • 5.3.15.81 por_hnf_cbusy_write_limit_ctl on page 804 • 5.3.15.82 por_hnf_cbusy_resp_ctl on page 805 • 5.3.15.83 por_hnf_cbusy_sn_ctl on page 806 • 5.3.15.95 por_hnf_pmu_mpam_sel on page 820 • 5.3.15.96 por_hnf_pmu_mpam_pardid_mask0-7 on page 822
Updated various HN-F register descriptions.	• 5.3.15.6 por_hnf_cfg_ctl on page 714 • 5.3.15.7 por_hnf_aux_ctl on page 716 • 5.3.15.8 por_hnf_r2_aux_ctl on page 720 • 5.3.15.9 por_hnf_cbusy_limit_ctl on page 722
Clarified a step in the boot-time programming requirements.	5.4.1 Boot-time programming sequence on page 837
Added PortID assignment register to CCIX-related programmable registers list.	5.5.1 CML-related programmable registers on page 849
Updated supported SN-F types and por_hnf_sam_sn_properties values to include CHI-C.	6.2.4 HN-F SAM configuration by SN type on page 870
Clarified that on software-configurable parity error injection, Data Check field of the DAT flit that is returned to an RN is altered.	6.3.2 Software-configurable parity error injection on page 888
Corrected the name of the *_mpam_override_en register field.	6.4.5.1 MPAM propagation on page 892
Clarified that MPAM is configured using configuration parameters and supported MPAM features are stored in por_hnf_mpam_idr.	6.4.5.2 MPAM configuration on page 894
Updated MPAM error case description to add interrupt signals that are triggered and exceptions to reporting behavior.	6.4.5.5 MPAM error logging and reporting on page 895
Added description of new CXRA performance monitoring events.	8.2 About the Performance Monitoring Unit on page 927
Added description of *Read Request Tracker* (RRT) slicing for RN-I and RN-D NUM_RD_REQ configurations of 128 and 256.	8.4.2.2 Read and write delays at RN-I bridges on page 935
Corrected number of RN-I PMU events.	8.4.3 RN-I PMU event summary on page 936
Added details of dual DAT/RSP feature and the impact on XP PMU events.	8.8 XP PMU event summary on page 947
Added GICD_DESTID signal description	B.13 Configuration input signals on page 987
Updated AXI/ACE-Lite master interface signals to add new signals, clarify that the AxID signal widths differ for HN-I and SBSX, and add description of calculating SBSX AxID signal widths.	• B.4.3 AXI/ACE-Lite master interface signals on page 969 • B.4.4 Calculating the SBSX AxID signal widths on page 973

Table C-3: Differences between issue 0100-01 and issue 0200-02

Change	Location
Updated product branding information.	Whole document
Updated description of supported node types and devices to add or clarify the following information: • HN-D components include DTC • Description of HN-T node type • New HN type, HN-P	2.1 About CMN-650 on page 20
Updated CCIX specification revision that is supported.	• 2.2 Compliance on page 21 • 4.11.8 CML CCIX Slave Agent support on page 163

Change	Location
Updated product features to describe the following changes: • New maximum mesh size of 10x10 • Larger number of local RN-F interfaces • Support for HN-P node type • Support for CCIX port-to-port forwarding Removed reference to 10-bit TxnID, since this information is covered by the *AMBA® 5 CHI Architecture Specification*.	2.3 Features on page 21
Updated product interface diagram to reflect newly supported device counts.	Figure 3-5: CMN-650 external interfaces on page 38
Updated list of HN-I instances with extra functionality to include HN-P.	3.6 System component selection on page 51
Updated details of mesh sizing and top-level configuration details with the following information: • Support for HN-P • New maximum mesh X and Y dimensions of 10 • Permitted values for DCS count. • New global parameters: ◦ *META_DATA_EN* ◦ *PORTFWD_EN* • New value ranges, default values, or both value ranges and default values for the following global parameters: ◦ *REQ_RSVDC_WIDTH* ◦ *REQ_ADDR_WIDTH* ◦ *NUM_REMOTE_RNF* ◦ *RNSAM_NUM_NONHASH_REGION* ◦ *RNSAM_NUM_QOS_REGIONS* ◦ *XY_OVERRIDE_CNT* • Updated permitted number of the following resources: ◦ RN-Fs ◦ RN-Is ◦ RN-Ds ◦ HN-Is ◦ HN-Fs • Removed requirement for even number of HN-Is if CAL is used and number of HN-Fs that is a power of 2.	3.7 Deciding on the size of the mesh on page 54

Change	Location
Updated device placement and configuration details with the following information: • Clarification of required *NUM_RD_BUF* parameter values in certain configurations. • New global parameters: ◦ RN-I and RN-D *NUM_PREALLOC_RD_BUF* ◦ RN-I and RN-D *ID_WIDTH* ◦ HN-F *SF_RN_ADD_VECTOR_WIDTH* ◦ HN-F *SF_MAX_RNF_PER_CLUSTER* ◦ All HN-P parameters ◦ CXHA *HA_PASS_BUFF_DEPTH* ◦ CXLA *PORTFWD_EN* ◦ CXLA *PORTFWD_NUM_DYNAMIC_TXBUF* ◦ CXLA *PORTFWD_NUM_STATIC_TXBUF* • New value ranges, default values, or both value ranges and default values for the following parameters: ◦ RN-I and RN-D *NUM_RD_BUF* ◦ HN-F *NUM_ENTRIES_POCQ* • Clarified for SBSX *SBSX_CMO_ON_AW* parameter that when enabled, CMOs are only sent on AW channel. • Updated description of the following CXHA parameters to clarify what they specify when passive buffer is either enabled or disabled: ◦ *HA_NUM_REQS* ◦ *HA_NUM_WRBUF*	Change superseded by later version.
Added details of product revisions for r1p1, r1p2, and r2p0 releases.	2.8 Product revisions on page 31
Updated structure of functional description chapter.	Some changes superseded by later versions. • 4.1 Clocks and resets on page 63 • 4.2 Power management on page 73 • 4.12 Discovery on page 167 • 4.13 Link layer on page 178 • 4.15 PCIe integration on page 181 • 4.17 Reliability, Availability, and Serviceability on page 184 • 4.18 Transaction handling on page 202 • 4.19 Processor events on page 213 • 4.20 Quality of Service on page 213 Some changes superseded in newer document version

Change	Location
Moved several sections under new Components and structural configuration section.	• 3.4 Components on page 39 • 4.14 Backward compatible RN-F support on page 180 • 3.10 Example system configurations on page 56 • 3.11 Example CML system configurations on page 60 • 4.11.7 CML Symmetric Multiprocessor support on page 163 • 4.11.8 CML CCIX Slave Agent support on page 163 • 4.10.1 Dual DAT and RSP channels on page 139 • 4.11.7 CML Symmetric Multiprocessor support on page 163 • 4.11.8 CML CCIX Slave Agent support on page 163 • 4.11.6 CCIX Port Aggregation Groups on page 162 • 4.11.10 CML credit requirements on page 165
Merged clocks and resets section into one, and moved various sections from elsewhere to the clocks and resets section.	• 4.1 Clocks and resets on page 63 • 4.1.5 High-level Clock Gating (HCG) on page 69 • 4.1.6 External Clock Controller (ExtCC) on page 70 • 4.1.7 CML clock management on page 71
Moved a section to power management section.	4.2.9 RN entry to and exit from snoop and DVM domains on page 85
Moved several sections under new network layer functions section.	Some changes superseded by later versions. • 4.3.1 Node ID mapping on page 88 • 2.4.1 Addressing capabilities on page 24 • 4.5 RN SAM on page 96 • 4.6 RA SAM on page 115 • 4.7 HN-F SAM on page 116 • 4.8 SAM support for CCIX Port Aggregation on page 126 • 4.5.2 SAM memory region size configuration on page 100 • 4.9 HN-I SAM on page 129 • 4.10.5 Default XY routing behavior on page 145 • 4.10.6 Non-XY routing on page 146
Moved several sections under new transaction handling section.	• 4.18.1 Atomics on page 202 • 4.18.2 Exclusive accesses on page 204 • 4.18.3 Barriers on page 206 • 4.18.4 Distributed Virtual Memory messages on page 206
Moved several programming tasks from the functional description chapter to the programmers model chapter.	• 5.4.4 Program the dual DAT/RSP channel selection scheme on page 843 • 5.4.6 RN-I and HN-I PCIe programming sequence on page 844 • 5.4.3 RN SAM and HN-F SAM programming on page 838 • 5.5.3.11 Program CPA functionality in RN SAM on page 860 • 5.5.3.12 Program CPA functionality in HN-F SAM on page 861 • 5.4.5 Program non-XY routing registers on page 843 • 5.4.7 DT programming on page 846 • 5.4.8 PMU system programming on page 847

Change	Location
Added new section describing dedicated RN-I resources AXI port traffic.	4.10.4 Dedicated RN-I resources for AXI port traffic on page 143
Added new section describing CXHA passive buffer support.	4.11.9 CXHA passive buffer support on page 164
Updated crosspoint description to increase the number of supported XPs and maximum mesh size.	3.2 Crosspoint (XP) on page 34
Added description of HN-P node.	Change superseded by later version.
Updated description of SAM component to indicate that RN SAM provides target IDs for HN-P.	4.4 System Address Map (SAM) on page 95
Updated description of CAL component to describe new CAL types that are supported.	3.4.6 Component Aggregation Layer (CAL) on page 46
Updated example asynchronous mesh topology figure to correct the locations of the individual clock domains.	Figure 3-10: Example asynchronous mesh topology on page 50
Removed restriction on the number of DCSs that are permitted when CCS is used.	3.4.7.4 CAL Credited Slice (CCS) on page 50
Corrected note describing condition under which DCT from CHI-B or CHI-C RN-F to CHI-D RN-F cannot be done. Updated HN-F backward compatibility for new fields.	4.14 Backward compatible RN-F support on page 180
Corrected dual DAT/RSP channel selection mechanism figure.	Figure 4-29: Dual DAT/RSP channel selection mechanism on page 141
Updated requirements for reset signal assertion and deassertion.	4.1.8 Reset on page 72
Added clarification about support for P-Channel initialization with **PREQ** asserted at **nSRESET** deassertion.	4.2.3 P-Channel on device reset on page 78
Updated node ID mapping scheme to describe changes for new maximum mesh size.	4.3.1 Node ID mapping on page 88
Added guidelines for CMN-650 CML systems that use *PA_WIDTH* of less than the minimum width that is supported by CCIX.	2.4.1 Addressing capabilities on page 24
Removed details of ROOTNODEBASE address, as it no longer applies to CMN-650.	4.12.1 Configuration address space organization on page 169
Updated discovery section to remove some information that was unclear. Clarified that node and configuration register offsets can be found in the IP-XACT file for your configuration.	• 4.12.1 Configuration address space organization on page 169 • 4.12.3 Child pointers on page 174
Updated description of DVM messages to remove a note describing some requirements relating to DVMs and RNs. Also added details of early DVM operation completion.	4.18.4 Distributed Virtual Memory messages on page 206
Added a PCIe topology restriction regarding HN-Ps.	4.15.1 PCIe topology requirements on page 181
Updated PCIe master and slave restrictions and requirements to include details of HN-P. Also added per-port reservation flow control requirements and information.	4.15.2 PCIe master and slave restrictions and requirements on page 181
Updated PCIe system requirements to include details of HN-P.	4.15.3 System requirements for PCIe devices on page 182
Added details of peer-to-peer transactions regarding HN-P, and made structural and editorial improvements.	5.4.6 RN-I and HN-I PCIe programming sequence on page 844
Reworked title of error handling section to Reliability, Availability, and Serviceability, and updated section to describe new ERRGSR register structure.	4.17 Reliability, Availability, and Serviceability on page 184
Updated information about request errors at HN-I to describe handling at HN-P.	4.17.6.1 Request errors at HN-I on page 195

Change	Location
Updated details of NDEs and DEs at HN-I to include HN-P handling of unsupported Exclusive accesses.	• 4.17.6.4 HN-I summary on sending NDE and DE on page 197 • 4.17.6.5 HN-I summary on logging errors on page 197
Updated description of RN-I error handling to remove information that is no longer applicable and added information about parity errors.	4.17.8 RN-I error handling on page 199
Updated XP configuration register flit fields table with 5-bit ERRSRC fields and values.	4.17.9 XP error handling on page 199
Updated CXHA error handling description to include details about metadata fields.	4.17.10 CXHA error handling on page 200
Updated description of RN-I and RN-D atomic request handling to describe RDB for atomic responses.	4.18.1.4 Atomic requests in RN-I and RN-D on page 203
Updated description of support for exclusive accesses in HN-I to clarify that the number of exclusive monitors supported in configuration-dependent. Also added note clarifying that HN-P does not support exclusive accesses.	4.18.2.2 Exclusive accesses in HN-I on page 204
Updated description of CML support for exclusive accesses to include non-SMP mode information.	4.18.2.3 CML support for exclusive accesses on page 204
Added section describing support for early completion of DVMOp requests.	4.18.4.1 Support for early completion of DVMOp requests on page 206
Added more detail about configurable threshold for measuring CBusy for the last 128 or 256 transactions.	4.18.5.1 Advanced CBusy handling in HN-F on page 208
Updated RN SAM target ID selection information to include HN-P and made some minor clarifications.	4.5.3 RN SAM target ID selection on page 103
Clarified memory map size alignment requirements.	Change superseded by later version.
Updated SCG description for configurations with 64 HN-Fs. Expanded example programming tables to include more scenarios and clarified how to distribute nodeIDs between multiple SCG. Added detail regarding non-hashed mode for SCGs.	4.5.4 System Cache Groups (SCGs) on page 104
Updated description of support for PrefetchTgt operations in RN SAM to include details of address range-based targets.	• 4.5.9 Support for PrefetchTgt operations in RN SAM on page 113 • 4.5.10 Address range-based QoS override and PrefetchTgt support on page 113
Updated section to: • Clarify that the address bit range that is used to compare against the mask register is based on the most significant address bit, rather than bit 51. Bit 51 might not be present in some configurations. • Correct guidelines around address bits when using 3-SN, 5-SN, or 6-SN mode. • Include 5-SN mode guidance for using address bit masking.	4.5.8 Address bit masking in the RN SAM on page 112
Various updates to HN-F with CAL support section: • Added configuration-dependent requirements for modifying specific RTL parameters when using HN-F CAL normal mode with large numbers of HN-Fs. • Added description of mixed CAL mode. • Updated limitations of this mode to include new 5-SN mode.	4.5.7 Support for HN-Fs with CAL in the RN SAM on page 110

Change	Location
Updated address-based QoS override section to describe relationship between address-based QoS override and PrefetchTgt features in the RN SAM. Also added more information about address range-based PrefetchTgt operations.	4.5.10 Address range-based QoS override and PrefetchTgt support on page 113
Added description of CXG targets that are permitted in each CPAG.	4.8 SAM support for CCIX Port Aggregation on page 126
Added details of new 5-SN address striping mode in HN-F SAM. Also clarified that direct-mapped mode supports 16 and 32 SNs.	4.7 HN-F SAM on page 116
Updated description of method to calculate PA bits that are stripped to include configurations with larger numbers of HN-Fs.	4.7.5 Maintaining contiguous address spaces in SN-Fs on page 124
Updated section to clarify that the address bit range that is used to compare against the mask register is based on the most significant address bit, rather than bit 51. Bit 51 might not be present in some configurations.	4.7.6 Address bit masking in the HN-F SAM on page 126
Made editorial updates and added context and prerequisites to the SAM programming task.	5.4.3.1 Program the SAM on page 839
Clarified that maximum addressable space is 2^{PA_WIDTH}.	4.5.2 SAM memory region size configuration on page 100
Made editorial updates and added context and prerequisites to the CPA functionality in RN SAM programming task. Also added details of new registers to program to set up CPA functionality in HN-F SAM.	• 5.5.3.11 Program CPA functionality in RN SAM on page 860 • 5.5.3.12 Program CPA functionality in HN-F SAM on page 861
Added note describing lack of support in HN-I of write streaming from RN-Fs.	4.9 HN-I SAM on page 129
Added note describing mapping of requests in HN-P.	4.9.3 HN-I SAM example configuration on page 131
In text, corrected default Y-axis routing directions when YIDs do not match.	4.10.5 Default XY routing behavior on page 145
Added section to describe extension to RAID mechanism to support large numbers of RN-Fs.	4.3.3 Extended CCIX Requesting Agent ID mechanism for up to 512 RN-Fs on page 92
Updated section to describe routing and ID mapping when either SF non-clustered or clustered mode is enabled. Added new requirements for this process. Removed 128 RN-F support section that was no longer applicable.	Change superseded by later version
Updated diagram describing RAID to LDID during CCIX discovery.	Figure 4-35: CXHA RAID to LDID LUT register format on page 154
Removed figure showing the number of RAID to LDID entries, as this information is also shown in Figure 4-35: CXHA RAID to LDID LUT register format on page 154.	Change superseded by later version
Added section describing CML port-to-port forwarding.	4.11.11 CML port-to-port forwarding on page 165
Added note clarifying that certain requirements that are described in this section only apply to two-chip configurations, and guidance for larger configurations. Also updated description of **PUB_DESTID** input.	4.16 Generic Interrupt Controller communication over AXI4-Stream ports on page 183
Added sections to describe REQ RSVDC and DAT RSVDC propagation.	• 4.18.6 REQ RSVDC propagation on page 212 • 4.18.7 DAT RSVDC propagation on page 212
Updated power management section to remove descriptions of clock management, as these concepts are now described elsewhere.	4.2 Power management on page 73
Clarified a statement regarding SMP system requirements, as they must be built using the same version of CMN-650.	4.11.7 CML Symmetric Multiprocessor support on page 163

Change	Location
Various updates to node configuration register address mapping information for larger mesh sizes, removed ROOTNODEBASE, and added more detail about the APB interface.	5.1.1 Node configuration register address mapping on page 223
Added more detail about the APB interface.	5.1.5 Requirements of configuration register reads and writes on page 225
Added an RN-D register description.	5.3.1.5 por_rnd_unit_info2 on page 247
Merged several RN-D register descriptions into single parameterized version.	• 5.3.1.8 por_rnd_s0-2_port_control on page 252 • 5.3.1.9 por_rnd_s0-2_mpam_control on page 253 • 5.3.1.10 por_rnd_s0-2_qos_control on page 254 • 5.3.1.11 por_rnd_s0-2_qos_lat_tgt on page 256 • 5.3.1.12 por_rnd_s0-2_qos_lat_scale on page 257 • 5.3.1.13 por_rnd_s0-2_qos_lat_range on page 259
Added fields to one RN-D register.	5.3.1.7 por_rnd_aux_ctl on page 250
Updated encodings for PMU events for RN-D.	5.3.1.14 por_rnd_pmu_event_sel on page 260
Merged several CXRA register descriptions into single parameterized version.	• 5.3.2.8 por_cxg_ra_sam_addr_region_reg0-7 on page 272 • 5.3.2.10 por_cxg_ra_agentid_to_linkid_reg0-7 on page 274 • 5.3.2.11 por_cxg_ra_rni_ldid_to_exp_raid_reg0-9 on page 276 • 5.3.2.12 por_cxg_ra_rnd_ldid_to_exp_raid_reg0-9 on page 277 • 5.3.2.13 por_cxg_ra_rnf_ldid_to_exp_raid_reg0-127 on page 278 • 5.3.2.14 por_cxg_ra_rnf_ldid_to_nodeid_reg0-127 on page 279
Added CXRA register.	5.3.2.15 por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg0-127 on page 281
Updated offset of a CXRA register.	5.3.2.9 por_cxg_ra_agentid_to_linkid_val on page 273
Updated fields of CXRA registers.	• 5.3.2.6 por_cxg_ra_aux_ctl on page 270 • 5.3.2.17 por_cxg_ra_cxprtcl_link0_ctl on page 283 • 5.3.2.19 por_cxg_ra_cxprtcl_link1_ctl on page 287 • 5.3.2.21 por_cxg_ra_cxprtcl_link2_ctl on page 291 • 5.3.2.16 por_cxg_ra_pmu_event_sel on page 282
Added an HN-I register.	5.3.3.22 por_hnp_pmu_event_sel on page 322
Updated fields of HN-I registers.	• 5.3.3.4 por_hni_unit_info on page 298 • 5.3.3.9 por_hni_cfg_ctl on page 305 • 5.3.3.10 por_hni_aux_ctl on page 306
Updated encodings for error source for HN-I.	• 5.3.3.15 por_hni_errmisc on page 312 • 5.3.3.20 por_hni_errmisc_NS on page 319
Added CXLA registers.	• 5.3.4.28 por_cxla_agentid_to_portid_reg0 on page 359 - 5.3.4.35 por_cxla_agentid_to_portid_reg7 on page 368 • 5.3.4.36 por_cxla_agentid_to_portid_val on page 369 • 5.3.4.37 por_cxla_portfwd_ctl on page 370 • 5.3.4.38 por_cxla_portfwd_status on page 371
Updated register reset value and fields for a CXLA register.	5.3.4.6 por_cxla_aux_ctl on page 329

Change	Location
Updated fields for various CXLA registers.	• 5.3.4.4 por_cxla_unit_info on page 327 • 5.3.4.39 por_cxla_pmu_event_sel on page 372
Merged several DN register descriptions into single parameterized version.	• 5.3.5.7 por_dn_vmf0-15_ctrl on page 384 • 5.3.5.8 por_dn_vmf0-15_rnf0 on page 385 • 5.3.5.12 por_dn_vmf0-15_rnd on page 389 • 5.3.5.13 por_dn_vmf0-15_cxra on page 390
Extended and merged several DN register descriptions into single parameterized version.	• 5.3.5.9 por_dn_vmf0-15_rnf1 on page 386 • 5.3.5.10 por_dn_vmf0-15_rnf2 on page 387 • 5.3.5.11 por_dn_vmf0-15_rnf3 on page 388
Updated fields of a DN register.	5.3.5.6 por_dn_aux_ctl on page 383
Updated fields of various CXHA registers.	• 5.3.6.2 por_cxg_ha_id on page 393 • 5.3.6.21 por_cxg_ha_cxprtcl_link0_ctl on page 416 • 5.3.6.23 por_cxg_ha_cxprtcl_link1_ctl on page 420 • 5.3.6.25 por_cxg_ha_cxprtcl_link2_ctl on page 423 • 5.3.6.31 por_cxg_ha_errmisc on page 432 • 5.3.6.36 por_cxg_ha_errmisc_NS on page 437
Updated register reset and fields of a CXHA register.	5.3.6.5 por_cxg_ha_aux_ctl on page 396
Added CXHA registers.	5.3.6.9 por_cxg_ha_unit_info2 on page 401
Added expanded instances for CXHA RAID to RN-F LDID mapping registers, up to 255. This register description replaces the por_cxg_ha_rnf_raid_to_ldid_reg{0-7} and por_cxg_ha_rnf_raid_to_ldid_val register descriptions.	5.3.6.19 por_cxg_ha_rnf_exp_raid_to_ldid_reg_0-255 on page 413
Updated offset of several CXHA registers.	• 5.3.6.10 por_cxg_ha_agentid_to_linkid_reg0 on page 401 - 5.3.6.17 por_cxg_ha_agentid_to_linkid_reg7 on page 411 • 5.3.6.18 por_cxg_ha_agentid_to_linkid_val on page 412
Merged several RN SAM register descriptions into single parameterized version.	• 5.3.7.5 non_hash_mem_region_reg0-19 on page 442 • 5.3.7.6 non_hash_tgt_nodeid0-4 on page 444 • 5.3.7.7 sys_cache_grp_region0-3 on page 445 • 5.3.7.8 sys_cache_grp_secondary_reg0-3 on page 446 • 5.3.7.14 sys_cache_grp_hn_nodeid_reg0-15 on page 454 • 5.3.7.15 sys_cache_grp_sn_nodeid_reg0-15 on page 455 • 5.3.7.19 sys_cache_grp_sn_sam_cfg0-1 on page 460 • 5.3.7.24 cml_port_aggr_grp0-4_add_mask on page 472 • 5.3.7.25 cml_port_aggr_grp_reg0-1 on page 472
Merged several RN SAM register descriptions into single parameterized version and updated fields to include sn_tgtid_override bit.	5.3.7.27 sam_qos_mem_region_reg0-7 on page 476
Added new RN SAM register descriptions.	• 5.3.7.28 sys_cache_grp_region0_sn_nodeid_reg0-15 on page 477 • 5.3.7.29 sys_cache_grp_region1_sn_nodeid_reg0-15 on page 479
Updated register field encodings of an RN SAM register to include details of 5-SN mode.	5.3.7.12 sys_cache_grp_sn_attr on page 451

Change	Location
Added various CFGM register descriptions. These register descriptions replace the previous por_cfgm_errgsr{0-7} and por_cfgm_errgsr{0-7}_NS registers.	• 5.3.9.10 por_cfgm_errgsr_mxp_0-7 on page 516 • 5.3.9.11 por_cfgm_errgsr_mxp_0-7_NS on page 517 • 5.3.9.12 por_cfgm_errgsr_hni_0-7 on page 518 • 5.3.9.13 por_cfgm_errgsr_hni_0-7_NS on page 518 • 5.3.9.14 por_cfgm_errgsr_hnf_0-7 on page 519 • 5.3.9.15 por_cfgm_errgsr_hnf_0-7_NS on page 520 • 5.3.9.16 por_cfgm_errgsr_sbsx_0-7 on page 521 • 5.3.9.17 por_cfgm_errgsr_sbsx_0-7_NS on page 522 • 5.3.9.18 por_cfgm_errgsr_cxg_0-7 on page 523 • 5.3.9.19 por_cfgm_errgsr_cxg_0-7_NS on page 523
Updated peripheral ID register encodings.	5.3.9.3 por_cfgm_periph_id_2_periph_id_3 on page 509
Updated fields of a CFGM register description.	5.3.9.29 por_info_global on page 532
Updated the title of a CFGM register description.	5.3.9.35 por_cfgm_child_pointer_0-255 on page 538
Updated fields of an SBSX register description.	• 5.3.11.1 por_sbsx_node_info on page 571 • 5.3.11.5 por_sbsx_cfg_ctl on page 576
Updated encodings for PMU events for SBSX.	5.3.11.18 por_sbsx_pmu_event_sel on page 590
Added RN-I register description.	5.3.13.5 por_rni_unit_info2 on page 634
Merged several RN-I register descriptions into single parameterized version.	• 5.3.13.9 por_rni_s0-2_mpam_control on page 640 • 5.3.13.10 por_rni_s0-2_qos_control on page 641 • 5.3.13.11 por_rni_s0-2_qos_lat_tgt on page 643 • 5.3.13.12 por_rni_s0-2_qos_lat_scale on page 644 • 5.3.13.13 por_rni_s0-2_qos_lat_range on page 646
Merged various RN-I register descriptions into single parameterized version and updated register fields.	5.3.13.8 por_rni_s0-2_port_control on page 639
Updated fields of an RN-I register description.	5.3.13.7 por_rni_aux_ctl on page 637
Updated encodings for PMU events for SBSX.	5.3.13.14 por_rni_pmu_event_sel on page 647

Change	Location
Merged several MXP register descriptions into single parameterized version.	• 5.3.14.2 por_mxp_device_port_connect_info_p0-1 on page 650 • 5.3.14.7 por_mxp_p0-1_info on page 657 • 5.3.14.13 por_mxp_p0-1_qos_control on page 664 • 5.3.14.14 por_mxp_p0-1_qos_lat_tgt on page 665 • 5.3.14.15 por_mxp_p0-1_qos_lat_scale on page 666 • 5.3.14.16 por_mxp_p0-1_qos_lat_range on page 667 • 5.3.14.22 por_mxp_p0-1_byte_par_err_inj on page 675 • 5.3.14.27 por_mxp_p0-1_syscoreq_ctl on page 681 • 5.3.14.28 por_mxp_p0-1_syscoack_status on page 683 • 5.3.14.31 por_dtm_fifo_entry0-3_0 on page 686 • 5.3.14.32 por_dtm_fifo_entry0-3_1 on page 687 • 5.3.14.33 por_dtm_fifo_entry0-3_2 on page 688 • 5.3.14.34 por_dtm_wp0-3_config on page 688 • 5.3.14.35 por_dtm_wp0-3_val on page 691 • 5.3.14.36 por_dtm_wp0-3_mask on page 692 • 5.3.14.42 por_mxp_multi_dat_rsp_chn_sel_0-15 on page 702 • 5.3.14.44 por_mxp_xy_override_sel_0-7 on page 705
Extended and merged a register description into one parameterized version.	• 5.3.14.6 por_mxp_child_pointer_0-31 on page 656
Extended and merged several MXP register descriptions into one parameterized version and updated relative_address_#{index} bit description.	5.3.14.6 por_mxp_child_pointer_0-31 on page 656
Updated MXP register fields.	Some changes superseded by future versions. • 5.3.14.9 por_mxp_secure_register_groups_override on page 659 • 5.3.14.21 por_mxp_errmisc on page 674 • 5.3.14.26 por_mxp_errmisc_NS on page 680 • 5.3.14.39 por_dtm_pmu_config on page 694
Added MXP register descriptions.	Some changes superseded by later version. • 5.3.14.11 por_mxp_p0-1_mpam_override on page 661 • 5.3.14.12 por_mxp_p0-1_ldid_override on page 662 • 5.3.14.31 por_dtm_fifo_entry0-3_0 on page 686-5.3.14.33 por_dtm_fifo_entry0-3_2 on page 688
Updated usage constraints and added secure group override for various MXP register descriptions.	5.3.14.43 por_mxp_multi_dat_rsp_chn_ctrl on page 704
Merged several MXP register descriptions into single parameterized version, updated usage constraints, and added secure group override for various MXP register descriptions.	5.3.14.44 por_mxp_xy_override_sel_0-7 on page 705

Change	Location
Added HN-F register descriptions.	- 5.3.15.5 por_hnf_unit_info_1 on page 713 - 5.3.15.60 por_hnf_cxg_ha_metadata_exclusion_list on page 777 - 5.3.15.97 por_hnf_rn_cluster0-63_physid_reg0 on page 823 - 5.3.15.98 por_hnf_rn_cluster64-127_physid_reg0 on page 826 - 5.3.15.99 por_hnf_rn_cluster0-127_physid_reg1 on page 829 - 5.3.15.100 por_hnf_rn_cluster0-127_physid_reg2 on page 831 - 5.3.15.101 por_hnf_rn_cluster0-127_physid_reg3 on page 834
Merged several HN-F register descriptions into single parameterized version.	5.3.15.96 por_hnf_pmu_mpam_pardid_mask0-7 on page 822
Removed por_hnf_rn_phys_id0-63 and por_hnf_ldid_map_table_reg0-4 register descriptions.	5.2.15 HN-F register summary on page 240
Updated fields of various HN-F register descriptions.	- 5.3.15.6 por_hnf_cfg_ctl on page 714 - 5.3.15.7 por_hnf_aux_ctl on page 716 - 5.3.15.63 por_hnf_sam_control on page 780 - 5.3.15.68 por_hnf_sam_sn_properties1 on page 788 - 5.3.15.94 por_hnf_pmu_event_sel on page 816
Made structural and editorial updates to various programming tasks, adding context and prerequisites where applicable.	- 5.4.1 Boot-time programming sequence on page 837 - 5.4.3.1 Program the SAM on page 839 - 5.5.4 Program CMN-650 CML system at runtime on page 863 - 5.5.5 Establish protocol link up between CXG and remote CCIX link on page 864 - 5.5.6 Link down CCIX protocol link between CXG and remote CCIX link on page 865
Clarified which registers need to be programmed, to indicate that the programming relates to WPN.	5.4.7.1 Program DTM watchpoint on page 846
Added a further step to DTC programming to enable cycle count.	5.4.7.2 Program DTC on page 847
Updated the list of CCIX-related programmable registers.	5.5.1 CML-related programmable registers on page 849
Divided CML bring up sequence flow into a top-level task and various subtasks. Also made structural and editorial updates to the programming steps, and added prerequisites and context.	- 5.5.2 Bring up a CML system on page 851 - 5.5.2.1 Discover and bring up a local CMN-650 system on page 851 - 5.5.2.2 Discover CCIX devices in CCIX system on page 852 - 5.5.2.3 Enumerate and configure CCIX devices on page 853

Change	Location
Divided CCIX communication programming steps into a top-level task, various subtasks, and supporting information. Also made structural and editorial updates to the programming steps, and added prerequisites and context.	• 5.5.3 Program CML system to enable CCIX communication on page 853 • 5.5.3.1 Options when programming CXG auxiliary control and configuration control registers on page 854 • 5.5.3.2 Assign IDs for local CXRAs and CXHAs on page 855 • 5.5.3.3 Assign LinkIDs to remote CCIX protocol links on page 856 • 5.5.3.5 Assign PCIe bus numbers for LinkIDs on page 857 • 5.5.3.6 Assign LDIDs to remote caching agents on page 857 • 5.5.3.7 Program RA SAM on page 858 • 5.5.3.8 Program RN SAM in CXHA on page 858 • 5.5.3.9 Program CCIX protocol link control registers on page 859 • 5.5.3.10 CCIX protocol link credit distribution on page 859
Added new CCIX programming task.	5.5.3.13 Enable CCIX port-to-port forwarding on page 862
Updated structure of SLC memory system chapter.	• 6.1 About the SLC memory system on page 867 • 6.2 SLC memory system components and configuration on page 869 • 6.3 Error reporting and software-configured error injection on page 887 • 6.4 Transaction handling in SLC memory system on page 888 • 6.5 QoS features on page 898
Moved several sections under new SLC memory system components and structural configuration section.	• 6.2.1 HN-F configurable options on page 869 • 6.2.2 Snoop connectivity and control on page 870 • 6.2.3 TrustZone technology support on page 870 • 6.2.4 HN-F SAM configuration by SN type on page 870 • 6.2.5 Hardware-based cache flush engine on page 871 • 6.2.6 Software-configurable memory region locking on page 873 • 6.2.7 Software-configurable On-Chip Memory on page 875 • 6.2.8 Source-based SLC cache partitioning on page 876 • 6.2.9 Way-based SLC cache partitioning on page 877
Moved several sections under new transaction handling in SLC memory system section.	• 6.4.1 Cache maintenance operations on page 888 • 6.4.2 Cacheable and non-cacheable exclusives on page 889 • 6.4.3 DataSource handling on page 889 • 6.4.4 CMO and PCMO propagation from HN-F to SN-F or SBSX on page 890 • 6.4.5 Memory System Performance Resource Partitioning and Monitoring on page 892
Updated HN-F configurable options information to describe options to set up SF clustered mode.	6.2.1 HN-F configurable options on page 869
Updated note describing the requirements of maximum number of logical processors concurrently sending exclusive requests, to add guidance for configurations with more than 64 RN-Fs.	6.4.2 Cacheable and non-cacheable exclusives on page 889

Change	Location
Updated description of multicast snoops to describe the relationship with SF clustered mode.	6.2.2 Snoop connectivity and control on page 870
Clarified that the valid bits for hnf_slc_lock_baseX registers depends on the value of the *PA_WIDTH* parameter.	Table 6-5: Settings for hnf_slc_lock_baseX on page 875
Updated information regarding calculating the variable memory region size.	6.2.6 Software-configurable memory region locking on page 873
Updated way-based cache partitioning example to describe the impact of SF clustered mode.	6.2.9 Way-based SLC cache partitioning on page 877
Corrected the error type that double-bit ECC error detection in SLC tag RAM belongs to, from uncorrectable error to deferred error.	6.3 Error reporting and software-configured error injection on page 887
Added new sections to describe SF clustered and non-clustered modes for RN-F tracking. Some changes superseded in later versions.	• 6.2.10 RN-F tracking in the SF on page 878 • 6.2.11 Non-clustered and clustered mode for SF RN-F tracking on page 879 • 6.2.12 Configuring clustered mode for SF tracking on page 881
Updated note describing HN-F tracking for logical processors to cover new maximum supported RN-F number. Updated guidelines for configurations with different numbers of RN-Fs.	6.4.2 Cacheable and non-cacheable exclusives on page 889
Updated MPAM propagation with information about software configurable MPAM override.	6.4.5.1 MPAM propagation on page 892
Added section describing software-programmable MPAM override.	6.4.5.6 Software-programmable MPAM override on page 896
Added information about maximum number of DTMs that are permitted in a single DTC domain and HN-T requirements for large mesh configurations.	• 7.1 Debug trace system overview on page 902 Some changes superseded by newer document version
Corrected DTM watchpoint comparator figure.	Figure 7-2: DTM WP comparator on page 904
Updated fields of various watch point match groups and trace data format.	• 7.1.1.1 WP match value and mask register on page 905 • 7.1.2.1 Trace data format on page 909
Updated description of read mode to clarify the behavior for specific packet types.	7.1.3 Read mode on page 912
Corrected alignment sync packet size details.	7.1.5.2 Alignment sync packet format on page 914
Clarified that the trace_tag_enable field of the por_dtm_control register must be set for trace tag to be generated.	• 7.2.2.1 Trace tag generation on page 919 • 7.2.2.4 Trace tag example programming on page 920
Updated the description of the example cross trigger sequence.	7.2.4.1 Cross trigger example programming on page 922
Added section describing CXLA PMU system.	7.4 CXLA PMU system on page 924
Added information regarding disabling read data interleaving on RN-I and RN-D.	8.1 Performance optimization guidelines on page 926
Added section describing RN-I and RN-D write burst cracking.	8.1.1 RN-I and RN-D write burst cracking on page 927
Added section describing snoop events related to SF clustering.	8.3.5 Snoop events related to SF clustering on page 931
Updated table of HN-F PMU events.	8.3.7 HN-F PMU event summary on page 931
Added section describing HN-P PMU event information.	8.6.4 HN-P PMU events on page 945
Added information about the conditions when the por_hnp_pmu_event_sel register sends output.	8.7 DN performance events on page 946
Added section describing CXG performance events.	8.9 CXG performance events on page 947

Change	Location
Updated value of Read_Interleaving_Disabled property support on RN-I. Also corrected value of DVM_v8 and DVM_v8.1 property, and DVM_v8.4 property for RN-I.	A.1 AXI and ACE-Lite feature support on page 950
Added details of extra global clock signals that are added when using multiple asynchronous clock domains.	B.2 Clock and reset signals on page 955
Various updates to ACE-Lite-with-DVM slave interface signal descriptions: • Added **AWCMO_S** signal. • Updated width and description of **WUSER_S** and **RUSER_S** signals. • Added width calculation for **WPOISON_S** and **RPOISON_S** signals.	B.4.2 ACE-Lite-with-DVM slave interface signals on page 964
Various updates to AXI/ACE-Lite interface signal descriptions: • Updated width and description of **WUSER_M** and **RUSER_M** signals. • Added width calculation for **AWUSER_M**, **AWID_M**, and **ARID_M** signals. • Added missing connection information for various signals.	B.4.3 AXI/ACE-Lite master interface signals on page 969
Updated and corrected calculations for **AxID** signals.	B.4.4 Calculating the SBSX AxID signal widths on page 973
Added new section.	B.4.5 HN-I and HN-P AxID signal properties and encodings on page 974

Table C-4: Differences between issue 0200-02 and issue 0200-03

Change	Location
Product branding changes to CMN-650 (previously known as CMN-Rhodes).	Whole document
Corrected number of supported RN-Is in list of features.	2.3 Features on page 21
Clarified the HN-I variants that are compatible with CAL.	3.8 Permitted numbers of devices and system resources in the mesh on page 55
Updated node ID mapping example figures to show decimal node ID values and added more textual detail to clarify how node IDs are calculated. Also clarified that bits [1:0] of the node ID in all formats correspond to the device ID, rather than a hard-coded `0b00`.	4.3.1 Node ID mapping on page 88
Corrected description of restrictions on SCG selection. Added description of separate SCG non-hashed mode register.	4.5.6 Configuring SCGs in the RN SAM on page 106
Added constraint that, when overriding RN-F LDIDs in SF clustered mode, values must match between por_cxg_ra_rnf_ldid_to_ovrd_ldid_reg_[0-127] and por_mxp_p[0-1]_ldid_override registers.	4.11.4 LDID assignment when using SF clustering on page 156
Updated RAS description to: • Clarify that the HN-D contains four sets of five error groups, rather than just five error groups • Update error interrupt handler flow example because the old example was incorrect	4.17 Reliability, Availability, and Serviceability on page 184

Change	Location
Updated description of HN-F to RN advanced CBusy handling to clarify that NonCopyBack Comp type requests do not include CMOs. Also corrected statement that previously said that atomic operations are not counted towards CopyBack or NonCopyBack type requests.	4.18.5.1 Advanced CBusy handling in HN-F on page 208
Corrected required Secure access **PPROT[1]** value for the APB slave port. **PPROT[1]** must be set to 0 for a Secure access to the APB slave port.	5.1.5 Requirements of configuration register reads and writes on page 225
Added note to clarify how to calculate logical RN-F bit vector in certain registers.	• 5.3.5.8 por_dn_vmf0-15_rnf0 on page 385 • 5.3.5.9 por_dn_vmf0-15_rnf1 on page 386 • 5.3.5.10 por_dn_vmf0-15_rnf2 on page 387 • 5.3.5.11 por_dn_vmf0-15_rnf3 on page 388
Added specific reset values for certain por_hnf_mpam_idr register bits.	5.3.12.3 por_hnf_mpam_idr on page 594
Clarified requirements for stalling or preventing transactions at RN-F ESAM interfaces until RN SAM programming is complete.	5.4.1 Boot-time programming sequence on page 837
Corrected value of the example programming of the region1_size field of the non_hash_mem_region_reg1 register, to provide the correct encoding for a 1GB memory region.	Table 5-505: RN SAM registers and programmed values on page 842
Corrected example HN-F SAM programming information. Previously, hn_cfg_sam_top_address_bit0 and hn_cfg_sam_top_address_bit1 were in the wrong order in the table.	Table 5-506: HN-F programming information on page 843
Corrected parameter name for dual DAT/RSP channel selection scheme.	5.4.4 Program the dual DAT/RSP channel selection scheme on page 843
Updated the programming sequence to set up PMU counters.	5.4.8.1 Set up PMU counters on page 847
Added new sections and updated existing sections describing the build configuration and programming of SF clustering schemes, with examples.	• 6.2.11 Non-clustered and clustered mode for SF RN-F tracking on page 879 • 6.2.12 Configuring clustered mode for SF tracking on page 881
Updated the software mechanism to enable region locking to remove bullet regarding overlocking.	6.2.6 Software-configurable memory region locking on page 873
Added note to clarify that CMN-650 does not support source-based or way-based SLC partitioning with MPAM.	6.4.5.3 Cache portion and capacity partitioning on page 894
Added details of configuration bit mechanism to trigger MPAM external capture event 6.	6.4.5.4 Cache capacity monitoring on page 895
Corrected the figure showing the trace data packet header format.	Figure 7-3: Trace data packet header on page 914
Clarified the order of the trace tag example sequence steps.	7.2.2.4 Trace tag example programming on page 920
Clarified the order of the flit tracing example sequence steps.	7.2.1.1 Flit tracing example on page 918
Clarified information that describes when trace packets are generated.	7.2.2.3 Trace tag trace packet generation on page 920
Updated description of function of secure_debug_disable field of por_dt_secure_access register, specifically clarifying that when the bit is set, Secure transactions are not traced.	7.2.4.2 Sample profile on page 923
Added new section describing the structure of the DAT.RSVDC field and subfields.	B.3.4 DAT.RSVDC subfields on page 959
Added description of MTU color when calculating the AxID signal widths.	B.4.4 Calculating the SBSX AxID signal widths on page 973

Change	Location
Corrected **SYSCOREQ_S** connection information.	Location of change superseded by future version.
The following changes are for improved content quality. The general technical meaning of the content has not changed.	
Added new components and configuration chapter, which contains some of the information from the introduction and functional description of the previous release.	3 Components and configuration on page 33
Updated description of external interfaces, figure, and table to include more information about the interfaces.	3.3 External interfaces on page 37
Updated components section to add summary tables of external devices, internal devices, and mesh components. Moved description of *CHI Domain Bridge* (CDB) and *AMBA Domain Bridge* (ADB) components to the mesh components table.	3.4 Components on page 39
Added description of interchangeable terms for XP components.	3.2 Crosspoint (XP) on page 34
Updated RN-I description to clarify information that is shared with or related to RN-D.	3.4.1 I/O coherent Request Node (RN-I) and I/O coherent Request Node with DVM support (RN-D) on page 42
Updated HN-I description to add more information about HN-I variants and condense information about colocated blocks.	3.4.3 I/O coherent Home Node (HN-I) on page 43
Updated structure of information about configuring CMN-650, to clarify the process. Also added more supporting information regarding the process.	3.5 Configure CMN-650 on page 50
Added more supporting information about factors to consider when selecting components.	3.6 System component selection on page 51
Created separate sections for: • Information on deciding on the size of the mesh • Information on permitted numbers of devices and system resources • Configurable options for mesh structure • Global configuration parameters. Also moved information regarding addressing capabilities alongside global configuration parameters. • Device-level configuration parameters. Also moved guidance on selecting the optimal total SF size across all HN-Fs to this section. Removed information about placing devices in the mesh that was no longer useful.	• 3.7 Deciding on the size of the mesh on page 54 • 3.8 Permitted numbers of devices and system resources in the mesh on page 55 • 3.9 Configurable options for mesh structure on page 56 • 2.4 Global configuration parameters on page 23 • 2.4.1 Addressing capabilities on page 24 • 2.5 Device-level configuration parameters on page 25
Added note describing specific clocking requirements for the P-Channel interface.	4.2.2 Power domain control on page 76
Added section describing *Logical Device IDs* (LDIDs).	4.3.2 Logical Device IDs on page 92
Condensed conceptual overview information about the SAM.	4.4 System Address Map (SAM) on page 95
Added more supporting high-level conceptual information about the RN SAM.	4.5 RN SAM on page 96

Change	Location
Created separate sections and more supporting information for the following: • Information about RN SAM memory regions, target types, and region requirements. Condensed conceptual information regarding memory regions from multiple separate locations. Added clarification about the handling of read and write requests targeting HN-D that do not target the configuration register space. • Condensed conceptual information regarding *System Cache Groups* (SCGs) from multiple separate locations. • Information about RN SAM target ID selection • Information about SCG HN-F hash algorithm • Information about configuring SCGs in the RN SAM. Also clarified that information about hashed target ID allocation in SCG target ID registers is also applicable to SN target IDs.	• 4.5.1 RN SAM memory regions and target types on page 97 • 4.5.3 RN SAM target ID selection on page 103 • 4.5.4 System Cache Groups (SCGs) on page 104 • 4.5.5 SCG HN-F hash algorithm on page 105 • 4.5.6 Configuring SCGs in the RN SAM on page 106
Divided tables showing RN SAM and HN-F SAM memory partition sizes for GIC regions and hashed/non-hashed regions for clarity.	• Table 4-12: RN SAM and HN-F SAM configuration register GIC memory region sizes on page 101 • Table 4-13: RN SAM and HN-F SAM configuration register hashed and non-hashed memory region sizes on page 101
Created separate section for information about RA SAM address region requirements.	4.6.1 RA SAM address region requirements on page 116
Added more supporting high-level information about the HN-F SAM.	4.7 HN-F SAM on page 116
Created separate sections and more supporting information for the following: • Information about mapping SN targets in the HN-F SAM • Information about HN-F SAM target ID selection • Examples of 3-SN, 5-SN, and 6-SN mode configurations	• 4.7.1 Mapping SN targets in the HN-F SAM on page 117 • 4.7.2 HN-F SAM target ID selection on page 118 • 4.7.4 Example 3-SN, 5-SN, and 6-SN mode configurations on page 120
Added more supporting high-level information about the HN-I SAM.	4.9 HN-I SAM on page 129
Created separate sections and more supporting information for: • Information about HN-I SAM address region 0 • Information about configuring HN-I SAM address regions and order regions	• 4.9.1 HN-I SAM address region 0 on page 130 • 4.9.2 Configuring HN-I SAM address regions and order regions on page 130
Updated description of HN-I SAM example configuration to clarify information about the configuration.	4.9.3 HN-I SAM example configuration on page 131
Created separate sections and more supporting information for: • Conceptual information about routing transactions across multi-chip systems and the resources that are used for that purposes • Information about mapping LDIDs to RAIDs in CXRA and CXHA • Information about mapping SF LDID vector values to snoop targets in HN-F • Information about LDID assignment when using SF clustering • Examples of cross-chip routing and mapping of IDs in those example scenarios	• 4.11.1 Routing transactions across multi-chip systems on page 152 • 4.11.2 Mapping LDIDs to RAIDs in CXRA and CXHA on page 153 • 4.11.3 Mapping SF LDID vector values to snoop targets in HN-F on page 155 • 4.11.4 LDID assignment when using SF clustering on page 156 • 4.11.5 Cross-chip routing examples on page 157
Condensed conceptual overview information about QoS regulators.	4.20.2.1 QoS regulators on page 214

Change	Location
Removed table describing PMU events. The events are described in the por_*_pmu_event_sel register description for each block.	8.2 About the Performance Monitoring Unit on page 927

Table C-5: Differences between issue 0200-03 and issue 0200-04

Change	Location
Corrected RN SAM target ID selection policy diagram. Now shows the correct hashed TgtID mux control, driven from the defined SCG regions.	Figure 4-18: RN SAM target ID selection policy on page 103
Reversed the ordering of bits in example HN-I SAM programming tables for consistency with the ordering of bits in register references.	• Table 4-34: Example por_hni_sam_addrregion0_cfg register programming on page 134 • Table 4-35: Example por_hni_sam_addrregion1_cfg register programming on page 135 • Table 4-36: Example por_hni_sam_addrregion2_cfg register programming on page 136 • Table 4-37: Example por_hni_sam_addrregion3_cfg register programming on page 138
Corrected the name of the parameter that specifies the number of flit buffers for flit uploads from RN-F or SN-F. The correct parameter name is *RXBUF_NUM_ENTRIES*.	4.13.2 Flit uploads from RN-F or SN-F on page 179
Reversed the ordering of bits in advanced CBusy handling tables for consistency with the ordering of bits in register references.	• Table 4-61: por_hnf_cbusy_limit_ctl register for CBusy thresholds (all requests or read types) on page 209 • Table 4-62: Register for CBusy thresholds (write requests) on page 209 • Table 4-64: por_hnf_cbusy_resp_ctl register for configuring CBusy value on responses on page 211 • Table 4-65: Register for identifying SN groups on page 212 • Table 4-66: por_hnf_cbusy_sn_ctl register for CBusy sampling control on page 212
Corrected HN-F CBusy value propagation according to programming table. Added sn_cbusy_prop_en and cbusy_highest_of_all_en values for combination hnf_adv_cbusy_mode_en = `0b1` and hnf_cbusy_rd_wr_types_en = `0b1`.	Table 4-63: HN-F CBusy value propagation according to programming on page 210
Added warning describing programming restrictions on POCQavailability and QoS classes.	• 4.20.2.4 HN-F QoS support on page 217 • 6.5.2 QoS class and POCQ resource availability on page 899
Removed details of some MXP registers that were not applicable.	5.2.15 HN-F register summary on page 240
Added a separate section describing ACE-Lite-without-DVM slave interface signals for an RN-I bridge. Previously, these signals were shown in the same section as the ACE-Lite-with-DVM slave interface signals for an RN-D bridge.	B.4.1 ACE-Lite-without-DVM slave interface signals on page 960
Added separate sections describing ATPG and MBIST signals. Previously, these signals were shown in the same section.	• B.8 ATPG interface signals on page 981 • B.9 MBIST interface signals on page 985
Added separate sections describing CML clock management signals. Previously, these signals were shown in the same section as the other clock management signals.	B.10.1 CML clock management signals on page 986