



Arm[®] Neoverse[™] CMN-700 Coherent Mesh Network

Revision: r3p1

Technical Reference Manual Addendum

Non-Confidential

Copyright © 2023 Arm Limited (or its affiliates).
All rights reserved.

Issue 01

108055_0301_01_en



Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual Addendum

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Release Information

Document history

Issue	Date	Confidentiality	Change
0301-01	19 May 2023	Non-Confidential	First release for r0p0 EAC

Proprietary Notice

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

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

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

This document may include technical inaccuracies or typographical errors.

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

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws

and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

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

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

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

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

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

Inclusive language commitment

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

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.....	8
1.1 Product revision status.....	8
1.2 Intended audience.....	8
1.3 Conventions.....	8
1.4 Useful resources.....	10
2. Super Home Node in CMN-700.....	12
2.1 HN-S configurable options.....	12
3. Functional Description.....	17
3.1 Backward compatible RN-F Support.....	17
3.2 PCIe reads.....	18
3.3 Power management.....	18
3.3.1 Power domains.....	18
3.3.2 Power domain control.....	20
3.3.3 HN-F Memory retention mode.....	22
3.3.4 HN-F power domains.....	22
3.3.5 HN-F RAM PCSM Interface.....	27
3.3.6 HN-F power domain completion interrupt.....	27
3.4 Network layer functions.....	28
3.4.1 System Address Map.....	28
3.4.2 RN SAM.....	29
3.4.3 HN-F SAM.....	54
3.4.4 SAM programming examples.....	70
3.4.5 Cross chip routing and ID mapping.....	79
3.5 Reliability, Availability, and Serviceability.....	85
3.5.1 Error Detection and Deferred Error values.....	88
3.5.2 HN-F error handling.....	88
3.6 Transaction handling.....	90
3.6.1 Atomics.....	90
3.6.2 Exclusive accesses.....	90
3.6.3 Completer Busy indication.....	91
3.6.4 StrongNC RSVDC handling.....	98

3.7 Quality of Service.....	98
3.7.1 Microarchitectural QoS support.....	99
3.8 HN-S.....	100
3.8.1 SAM support for HN-S.....	101
3.8.2 LCN SAM.....	108
3.8.3 HN-S Completer Busy Indication.....	110
3.8.4 HN-S SLC and LCC capacity partitioning.....	110
3.8.5 Configuring non-clustered RN-F tracking in HN-S SF.....	111
3.8.6 HN-S error handling.....	111
3.8.7 HN-S POCQ Resource Allocation.....	112
3.8.8 HN-S restrictions.....	112
3.9 HN-F performance events.....	113
3.9.1 Cache performance.....	113
3.9.2 HN-F counters.....	114
3.9.3 SF events.....	115
3.9.4 System-wide events.....	116
3.9.5 Snoop events related to SF clustering.....	116
3.9.6 Quality of Service.....	117
3.9.7 HN-F PMU event summary.....	117
4. SLC memory system.....	120
4.1 About the SLC memory system.....	120
4.2 SLC memory system components and configuration.....	122
4.2.1 HN-F configurable options.....	123
4.2.2 Snoop connectivity and control.....	123
4.2.3 Realm Management Extension support.....	124
4.2.4 HN-F SAM configuration by SN type.....	124
4.2.5 Memory address decode error handling.....	125
4.2.6 Hardware-based cache flush engine.....	126
4.2.7 Software-configurable memory region locking.....	128
4.2.8 Software-configurable On-Chip Memory.....	130
4.2.9 Source-based SLC cache partitioning.....	131
4.2.10 Way-based SLC cache partitioning.....	132
4.2.11 RN-F tracking in the SF.....	134
4.2.12 Non-clustered and clustered modes for SF RN-F tracking.....	135
4.2.13 Configuring non-clustered RN-F tracking in HN-F SF.....	135

4.2.14 Configuring clustered mode for SF tracking.....	136
4.2.15 Identifying clusters and individual devices in clustered mode.....	137
4.3 Error reporting and software-configured error injection.....	143
4.3.1 Software-configurable error injection.....	144
4.3.2 Software-configurable parity error injection.....	144
4.4 Transaction handling in SLC memory system.....	144
4.4.1 Cache maintenance operations.....	144
4.4.2 Cacheable and Non-cacheable exclusives.....	145
4.4.3 DataSource handling.....	146
4.4.4 CMO and PCMO propagation from HN-F to SN-F or SBSX.....	147
4.4.5 Memory System Performance Resource Partitioning and Monitoring.....	148
4.4.6 MTE support in HN-F.....	155
4.5 HN-F class-based resource allocation and arbitration.....	155
4.5.1 Class assignment.....	155
4.5.2 POCQ resource allocation.....	156
4.5.3 POCQ request arbitration.....	161
5. Programmer's Model.....	163
5.1 Register Summary.....	163
5.1.1 Configuration register summary.....	163
5.1.2 HN-S MPAM_S register summary.....	164
5.1.3 HN-S MPAM_NS register summary.....	166
5.1.4 HN-S register summary.....	168
5.2 Register Description.....	174
5.2.1 Configuration register descriptions.....	174
5.2.2 HN-S MPAM_S register descriptions.....	186
5.2.3 HN-S MPAM_NS register descriptions.....	263
5.2.4 HN-S register descriptions.....	340
A. Revisions.....	564

1. Introduction

1.1 Product revision status

The r_xp_y identifier indicates the revision status of the product described in this manual, for example, $r1p2$, where:

r_x	Identifies the major revision of the product, for example, $r1$.
p_y	Identifies the minor revision or modification status of the product, for example, $p2$.

1.2 Intended audience

This book is written for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC) that uses the *Super Home Node* (HN-S) device type from the Arm® Neoverse™ CMN-700 Coherent Mesh Network interconnect.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

Convention	Use
<i>italic</i>	Citations.
bold	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	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: <pre>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></pre>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, IMPLEMENTATION DEFINED , IMPLEMENTATION SPECIFIC , UNKNOWN , and UNPREDICTABLE .



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



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



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



An important piece of information that needs your attention.



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



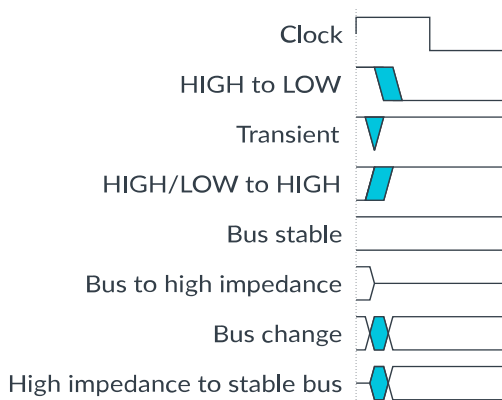
A reminder of something important that relates to the information you are reading.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

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

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

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

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

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

1.4 Useful resources

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

Access to Arm documents depends on their confidentiality:

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

Arm product resources	Document ID	Confidentiality
Arm® Neoverse™ CMN-700 Coherent Mesh Network Technical Reference Manual	102308	Non-Confidential
Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual	102309	Confidential

Arm product resources	Document ID	Confidentiality
Arm® Socrates™ User Guide	101399	Non-Confidential
Arm® Socrates™ Installation Guide	101400	Non-Confidential
Arm® Neoverse™ N1 hyperscale reference design GIC-600 Integration using CMN-600 AXI4-Stream Interfaces White Paper	PJDOC-1779577084-5931	Confidential
Arm® Neoverse™ CMN-700 Coherent Mesh Network Release Note	PJDOC-1779577084-33602	Confidential

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

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



Note

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

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

2. Super Home Node in CMN-700

CMN-700 is an AMBA 5 CHI interconnect with a customizable mesh topology and offers support to the *Super Home Node* (HN-S) device type.



This documentation details the *Super Home Node* (HN-S), which is only available through specific configurations and requires consent from Arm® to implement.



CMN-700 does not support a mixed configuration of HN-F and HN-S device types.

HN-S is a device that acts as a *Fully Coherent Home Node* (HN-F) for a local coherent memory region and as a *Local Coherency Node* (LCN) for a remote coherent memory region. HN-Ss accepts coherent requests from the RN-Fs and RN-Is, and generates snoops to all applicable *Request Node Full* (RN-F) in the system as required to support the coherency protocol.



Any descriptions related to the *Fully coherent Home Node* (HN-F) also apply to the HN-S.

CMN provides a high-performance distributed SLC and *Snoop Filter* (SF) with up to 128 HN-Fs and Cache sizes of 0 - 512 MB total:

- The HN-F includes an integrated *Point-of-Serialization* (PoS) and *Point-of-coherency* (PoC). The HN-F SLC (also referred to as an Agile System Cache) can be used for both computing and I/O caching
- The SF has up to 1024 MB of tag RAM for increased coherency scalability and consists of up to 128 partitions (with one for each HN-F).

2.1 HN-S configurable options

CMN-700 currently supports the following configurations using the HN-S device.



CMN does not support a mixed configuration of HN-F and HN-S device types.

The following table shows top-level configurable options specifically used for HN-S/HN-F device types:

Table 2-1: Top-level configurable options

Feature	Parameter	Description	Values	Comments
System Cache	The number of HN-S devices	The total number of HN-S instances in the system.	1-64 without CAL 2-128 with CAL	<p>When CAL is present, the number of HN-S devices must be even.</p> <p>The number of HN-S devices referred to by a given cache group must be a power of two.</p> <p>Note: CMN does not support a mixed configuration of HN-F and HN-S device types.</p>

The following table shows CHI device configurable options specifically used for HN-S/HN-F device types:

Table 2-2: CHI device configurable options

Parameter	Description	Values (Default)	Comments
SLC_SIZE	Size of system cache per HN-F node.	0K, 128K, 256K, 384K, 512K, 1M, 1.5M, 2M, 3M, 4M	<p>Valid SLC_SIZE, SLC_NUM_WAYS combinations are</p> <p>0K (SLC_SIZE = -8, NUM_WAYS = 16)</p> <p>128K (SLC_SIZE = -2, NUM_WAYS = 16)</p> <p>256K (SLC_SIZE = -1, NUM_WAYS = 16)</p> <p>384K (SLC_SIZE = 0, NUM_WAYS = 12)</p> <p>512K (SLC_SIZE = 0, NUM_WAYS = 16)</p> <p>1M (SLC_SIZE = 1, NUM_WAYS = 16)</p> <p>1.5M (SLC_SIZE = 2, NUM_WAYS = 12)</p> <p>2M (SLC_SIZE = 2, NUM_WAYS = 16)</p> <p>3M (SLC_SIZE = 3, NUM_WAYS = 12)</p> <p>4M (SLC_SIZE = 3, NUM_WAYS = 16)</p>
SF_SIZE	Size of SF tag RAM	512K, 1M, 2M, 4M, 8M, 16M	<p>Size of the SF tag RAM is chosen based on SF_NUM_WAYS as: $SF_SIZE = (32K * (2^{SF_SIZE_PARAM}) * SF_NUM_WAYS)$</p> <ul style="list-style-type: none"> SF_NUM_WAYS = 16: 512K, 1M, 2M, 4M, 8M SF_NUM_WAYS = 32: 1M, 2M, 4M, 8M, 16M

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

Parameter	Description	Values (Default)	Comments
HNS_MPAM_NS_NUM_CSUMON	Number of Non-Secure CSU monitoring counters	4, 8, 16, 32 (8)	-
HNS_MPAM_S_NUM_CSUMON	Number of Secure CSU monitoring counters	1, 2, 4 (2)	-

¹ *Point-of-Coherency Queue (POCQ)*

3. Functional Description

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



This documentation details the *Super Home Node* (HN-S), which is only available through specific configurations and requires consent from Arm® to implement.



CMN-700 does not support a mixed configuration of HN-F and HN-S device types.



Any descriptions related to the *Fully coherent Home Node* (HN-F) also apply to the HN-S.

3.1 Backward compatible RN-F Support

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

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

Table 3-1: HN-F backward compatibility

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

3.2 PCIe reads

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

If the remote target is an HN-F, the remote CCG RN-I cracks the read bursts into 64B chunks.



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

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

3.3 Power management

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

CMN-700 has the following power management capabilities:

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

These power states reduce static and dynamic power consumption.

- Support for static retention in HN-F in which the SoC places SLC and SF RAMs in a retention state. This capability reduces static power consumption.
- Support for in-pipeline low-latency Data RAM retention control, in which a programmable idle counter can be used to put the SLC RAMs in retention.

3.3.1 Power domains

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

The power domains are:

Logic

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

System Level Cache RAM0

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

System Level Cache RAM1

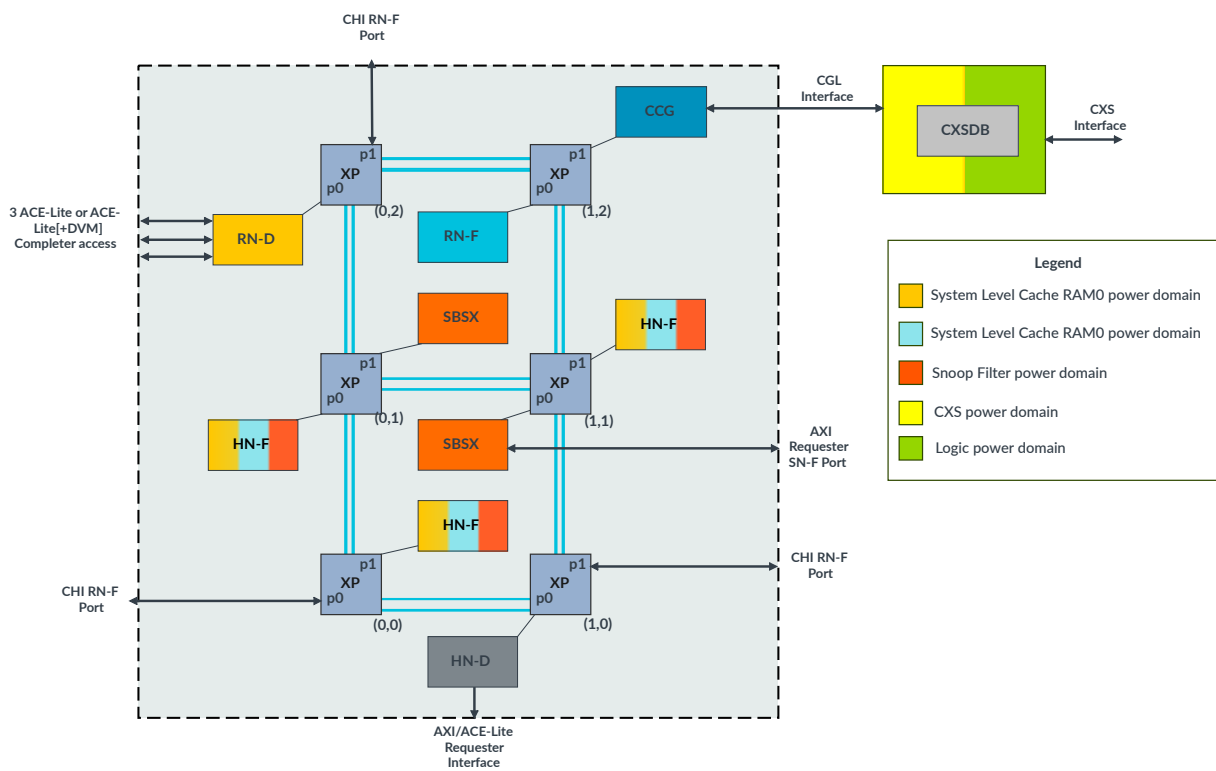
SLC Tag and Data RAMs way[15:8] within HN-F partitions. The RAMs in each HN-F partition can be independently controlled. The RAM1 domain for 384K, 1.5MB or 3MB SLC size configurations includes way[11:8].

Snoop filter only mode

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

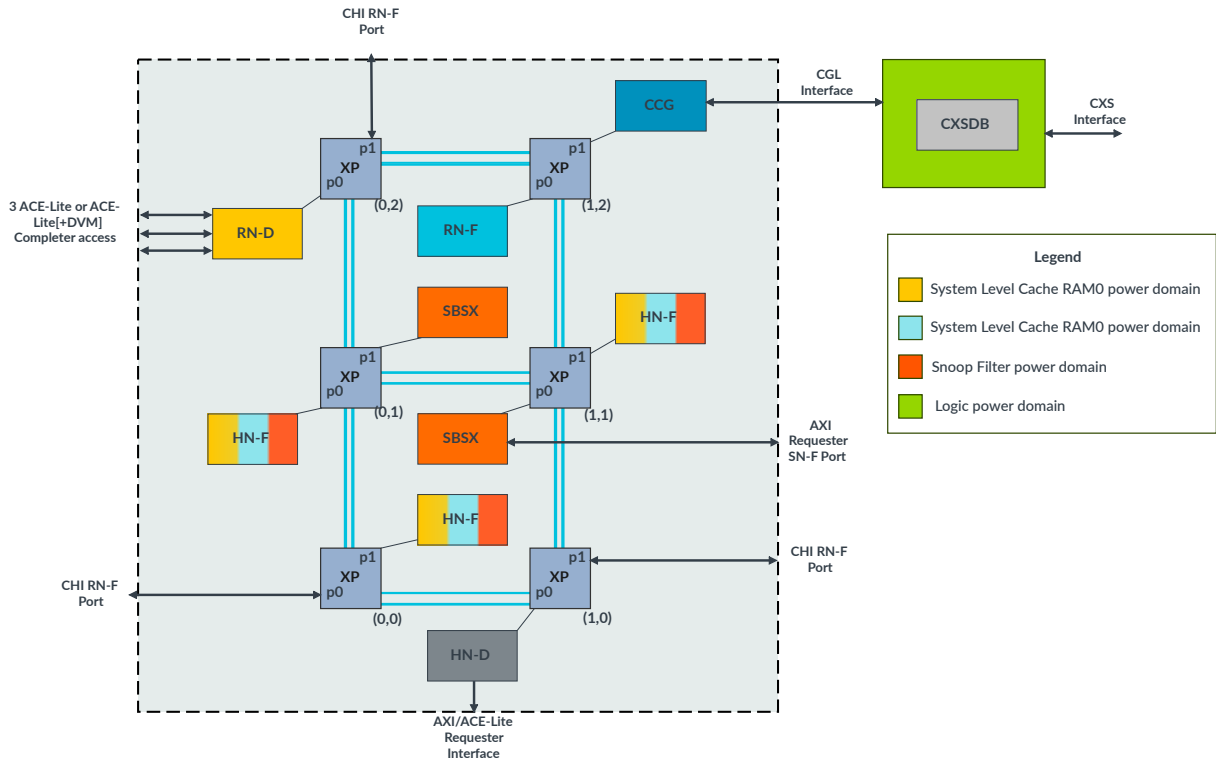
The following figure shows an example power domain configuration.

Figure 3-1: CMN-700 power domain example



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

Figure 3-2: Single CML power domain example

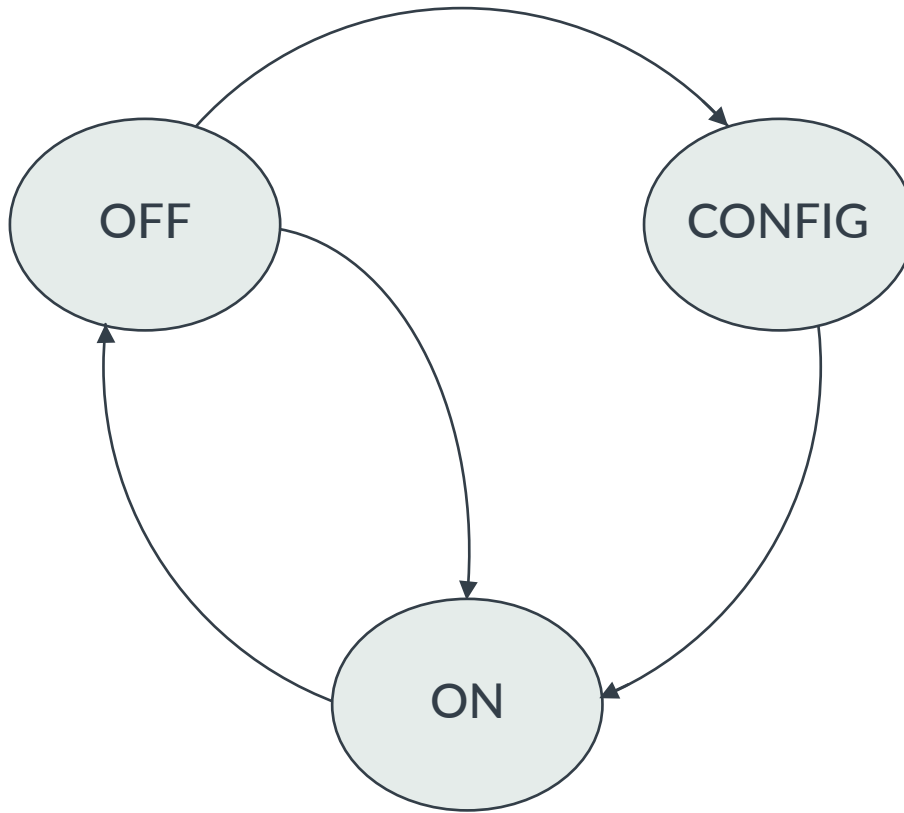


3.3.2 Power domain control

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

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

Figure 3-3: Logic domain states



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

Cold reset

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

Exit from HN-F Static Retention state

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

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

Table 3-2: Power mode configurations and PSTATE values

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

For an introduction to HN-F states, see [3.3.4 HN-F power domains](#) on page 22.

For P-Channel signal list information, see .

3.3.3 HN-F Memory retention mode

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

Entering HN-F Memory retention mode

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

Exiting HN-F Memory retention mode

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

3.3.4 HN-F power domains

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

The HN-F has three classes of power states:

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

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

FAM

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

HAM

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

SFONLY

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

NOSFSLC

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

The following constraints apply to the power states and transitions:

- If SLC size is 0KB, the HN-F does not support transitions to FAM or HAM modes
- After initialization, the power status register indicates FAM instead of SFONLY for 0KB SLC configurations
- When a power transition is initiated, another must not be initiated until the first one completes

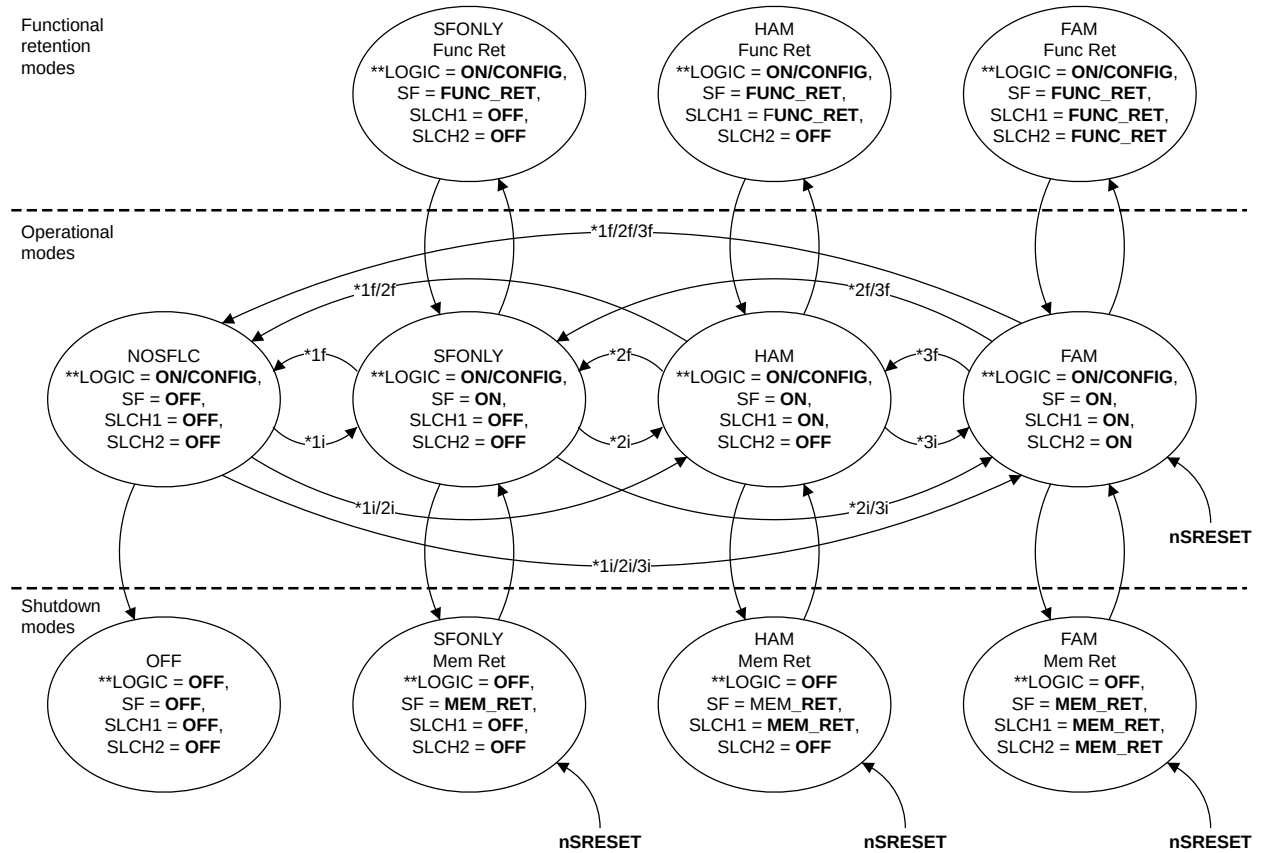
The following table shows the valid HN-F power states and their requirements.

Table 3-3: HN-F power states

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

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

Figure 3-4: Power state transitions



Note: **BOLD** text shows the required power state.

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

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

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

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



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

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

- `Policy`
- `op_mode`

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

- `pow_status`
- `op_mode_status`

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

From the FAM, HAM, or SFONLY modes, the HN-F can enter a dynamic retention mode using configuration register writes, where:

- The logic power is on
- The voltage to the RAMs is on, but is reduced to a level that is sufficient for bitcell retention but insufficient for normal operation
- The array pipeline is blocked, and a handshake occurs to allow array access when exiting the retention state

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

From these states, the SLC can also enter a Memory retention mode, where:

- The logic power is turned off
- The voltage to the RAMs is on, but is reduced to a level that is sufficient for bitcell retention but insufficient for normal operation
- Reset deassertion is essential when exiting retention after logic power down

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

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

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

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



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

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



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

Table 3-4: Power modes, operational modes, and RAM configurations

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

3.3.5 HN-F RAM PCSM Interface

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

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

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

Table 3-5: PSTATE Encodings

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



This interface does not have a PDENY signal.

3.3.6 HN-F power domain completion interrupt

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

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

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

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

3.4 Network layer functions

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

3.4.1 System Address Map

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

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

I/O space

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

DDR space

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



Unmapped addresses are routed to the HN-D.

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

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

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

RN SAM

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

CML RA SAM

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

HN-F SAM

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

HN-I SAM

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

3.4.2 RN SAM

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

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

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

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

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

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

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

The RN SAM programming sequence is described in .

For programming examples of different memory map configurations in the RNSAM and HNSAM, see [3.4.4 SAM programming examples](#) on page 70.

3.4.2.1 Hashed Target Groups

Hashed Target Groups (HTGs) are configured to support:

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

SCGs

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

The lowest four HTGs, HTGs 0-3, are only used for SCGs. This restriction preserves backward compatibility with legacy CMN products. The remaining HTGs can be used either for SCGs, or for HN-P or CCG PCIe hashing. SCGs support a maximum of 128 HN-F target IDs (or 256 total HN-Fs

with CAL2). You can configure the HN-F target IDs into one SCG or distribute them across multiple SCGs.

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

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

SCG: Power of two hashing, legacy CMN mode

Power of two hashing supports hashing over 1, 2, 4, 8, 16, 32, 64, 128, or 256 HN-Fs. The hash function uses bits [MSB:6] of the PA of the request. To use power of two hashing over 256 HN-F nodes, you must enable CAL mode. For more information, see [3.4.2.3 HN-F with CAL support](#) on page 41.

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

- Two HN-Fs:
 - Number of bits in select: 1
 - $\text{select}[0] = (6 \wedge 7 \wedge 8 \wedge \dots \wedge 51)$
- Four HN-Fs:
 - Number of bits in select: 2
 - $\text{select}[0] = (6 \wedge 8 \wedge 10 \wedge \dots \wedge 50)$
 - $\text{select}[1] = (7 \wedge 9 \wedge 11 \wedge \dots \wedge 51)$
- Eight HN-Fs:
 - Number of bits in select: 3
 - $\text{select}[0] = (6 \wedge 9 \wedge 12 \wedge \dots \wedge 51)$
 - $\text{select}[1] = (7 \wedge 10 \wedge 13 \wedge \dots \wedge 49)$
 - $\text{select}[2] = (8 \wedge 11 \wedge 14 \wedge \dots \wedge 50)$
- 16 HN-Fs:
 - Number of bits in select: 4
 - $\text{select}[0] = (6 \wedge 10 \wedge 14 \wedge \dots \wedge 50)$
 - $\text{select}[1] = (7 \wedge 11 \wedge 15 \wedge \dots \wedge 51)$
 - $\text{select}[2] = (8 \wedge 12 \wedge 16 \wedge \dots \wedge 48)$

- $\text{select}[3] = (9^{13} 17^{\dots} 49)$
- 32 HN-Fs:
 - Number of bits in select: 5
 - $\text{select}[0] = (6^{11} 16^{\dots} 51)$
 - $\text{select}[1] = (7^{12} 17^{\dots} 47)$
 - $\text{select}[2] = (8^{13} 18^{\dots} 48)$
 - $\text{select}[3] = (9^{14} 19^{\dots} 49)$
 - $\text{select}[4] = (10^{15} 20^{\dots} 50)$
- 64 HN-Fs:
 - Number of bits in select: 6
 - $\text{select}[0] = (6^{12} 18^{\dots} 48)$
 - $\text{select}[1] = (7^{13} 19^{\dots} 49)$
 - $\text{select}[2] = (8^{14} 20^{\dots} 50)$
 - $\text{select}[3] = (9^{15} 21^{\dots} 51)$
 - $\text{select}[4] = (10^{16} 22^{\dots} 46)$
 - $\text{select}[5] = (11^{17} 23^{\dots} 47)$
- 128 HN-Fs:
 - Number of bits in select: 7
 - $\text{select}[0] = (6^{13} 20^{\dots} 48)$
 - $\text{select}[1] = (7^{14} 21^{\dots} 49)$
 - $\text{select}[2] = (8^{15} 22^{\dots} 50)$
 - $\text{select}[3] = (9^{16} 23^{\dots} 51)$
 - $\text{select}[4] = (10^{17} 24^{\dots} 45)$
 - $\text{select}[5] = (11^{18} 25^{\dots} 46)$
 - $\text{select}[6] = (12^{19} 26^{\dots} 47)$
- 256 HN-Fs:
 - Number of bits in select: 8
 - $\text{select}[0] = (6^{14} 22^{\dots} 46)$
 - $\text{select}[1] = (7^{15} 23^{\dots} 47)$
 - $\text{select}[2] = (8^{16} 24^{\dots} 48)$
 - $\text{select}[3] = (9^{17} 25^{\dots} 49)$
 - $\text{select}[4] = (10^{18} 26^{\dots} 50)$
 - $\text{select}[5] = (11^{19} 27^{\dots} 51)$
 - $\text{select}[6] = (12^{20} 28^{\dots} 44)$
 - $\text{select}[7] = (13^{21} 29^{\dots} 45)$

SCG: Hierarchical hashing

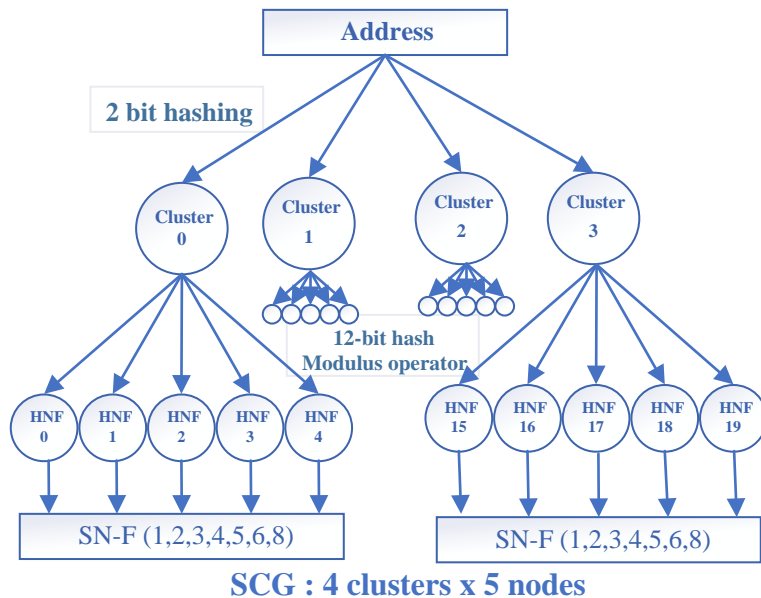
If you use hierarchical hashing, HN-Fs are grouped hierarchically in two levels. In the first level of hierarchy, HN-Fs are grouped into clusters and the addresses are hashed across the clusters. In the second level of hierarchy, addresses are hashed across the HN-F nodes in a cluster. To enable hierarchical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

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

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

In hierarchical hashing, the physical memory address that is hashed at the first level of hierarchy is sent to a single cluster within the SCG. Memory controllers downstream of HN-Fs must be configured to support either direct mapping per cluster or direct mapping per SCG. When using hierarchical hashing, direct SN mapping per HN-F is not supported. Also, the hierarchical hashing scheme can cause unused sets in the HN-F SLC and SF in certain use cases. For more information about mitigating unused sets, see [3.4.3.4 HN-F SLC and SF flexible addressing](#) on page 67.

Figure 3-5: Hierarchical hashing configuration



Non-power of two hash function

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

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

$\log_2(\text{number_of_clusters})$ to zero the LSBs $\text{Addr}[n:6]$ to achieve better HN-F and SLC set-ID distribution. For physical address spaces less than 52 bits, the upper bits are treated as zeros. The resulting address can be referred to as $\text{Addr}'[51:6]$.

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

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

Where num_hnf is either:

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

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

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

$$\text{offset} = (\text{first-level_result} * \text{hnfs_per_cluster}) + \text{select}[6:0]$$

When CAL mode is enabled for an SCG, the $\text{select}[6:0]$ result is used differently to determine the table index of the target HN-F node, and to modify the table entry to generate the final HN-F target ID value.. For more information about CAL mode, see [3.4.2.3 HN-F with CAL support](#) on page 41

SCG: User-defined hashing logic

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

Secondary memory regions for HTGs

Each HTG has two sets of programmable address regions and supports two memory regions. The HTGs can be programmed with two non-consecutive spaces.

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

SCG: target ID table configuration

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

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

- Flexible target ID base indexes (`RNSAM-FLEX-TGTID-EN = 1` (enabled by default)). At reset, all the SCG table base indexes point to zero. The table base index of each SCG is derived in a linked list approach, based on the SCG programming:
$$\text{SCG}[n] \text{ base index} = (\text{SCG}[n-1] \text{ base index}) + (\text{SCG}[n-1] \text{ number of HN-Fs})$$

For the following configuration types, this mechanism must be used and the `RNSAM-FLEX-TGTID-EN` parameter must be set:

- Configurations with more than four SCGs or HTGs (`RNSAM-NUM-HTG` parameter > 4)
- Configurations with more than 64 HN-F nodes

Figure 3-6: Example of HN-F flexible target ID configuration

HN-F Hashed Tgt-ID table	
HTG0-HN-F base index=0 HTG0 : num_HN-F configured = 16	Hashed Tgt-ID (0)
	Hashed Tgt-ID (1)
	Hashed Tgt-ID (2)
	Hashed Tgt-ID (3)
	Hashed Tgt-ID (...)
HTG1-HN-F base index=16 HTG1 : num_HN-F configured = 8	Hashed Tgt-ID (16)
	Hashed Tgt-ID (17)
	Hashed Tgt-ID (...)
HTG2-HN-F base index=24	Hashed Tgt-ID (24)
	Hashed Tgt-ID (...)
	Hashed Tgt-ID (NUM_HNF-1)

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

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

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

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

The hashing input is defined as:

$\text{rnsam_axid_in_port} = \{\text{port_id}, \text{AXID}\} = \{2 \text{ bits}, 32 \text{ bits}\} = 34 \text{ bits}$

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

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

- Two HN-Ps or CCGs in CPAG Hashing:
 - Number of bits in select: 1
 - $\text{select}[0] = (0^1 1^2 \dots ^{33})$
- Four HN-Ps or CCGs CPAG Hashing:
 - Number of bits in select: 2
 - $\text{select}[0] = (0^2 2^4 \dots ^{32})$
 - $\text{select}[1] = (1^3 3^5 \dots ^{33})$
- Eight HN-Ps or CCGs CPAG Hashing:
 - Number of bits in select: 3
 - $\text{select}[0] = (0^3 3^6 \dots ^{33})$
 - $\text{select}[1] = (1^4 4^7 \dots ^{31})$
 - $\text{select}[2] = (2^5 5^8 \dots ^{32})$
- 16 HN-Ps or CCGs CPAG Hashing:
 - Number of bits in select: 4
 - $\text{select}[0] = (0^4 4^8 \dots ^{32})$
 - $\text{select}[1] = (1^5 5^9 \dots ^{33})$
 - $\text{select}[2] = (2^6 6^{10} \dots ^{30})$
 - $\text{select}[3] = (3^7 7^{11} \dots ^{31})$
- 32 HN-Ps or CCGs CPAG Hashing:
 - Number of bits in select: 5
 - $\text{select}[0] = (0^5 5^{10} \dots ^{30})$
 - $\text{select}[1] = (1^6 6^{11} \dots ^{31})$
 - $\text{select}[2] = (2^7 7^{12} \dots ^{32})$
 - $\text{select}[3] = (3^8 8^{13} \dots ^{33})$
 - $\text{select}[4] = (4^9 9^{14} \dots ^{29})$

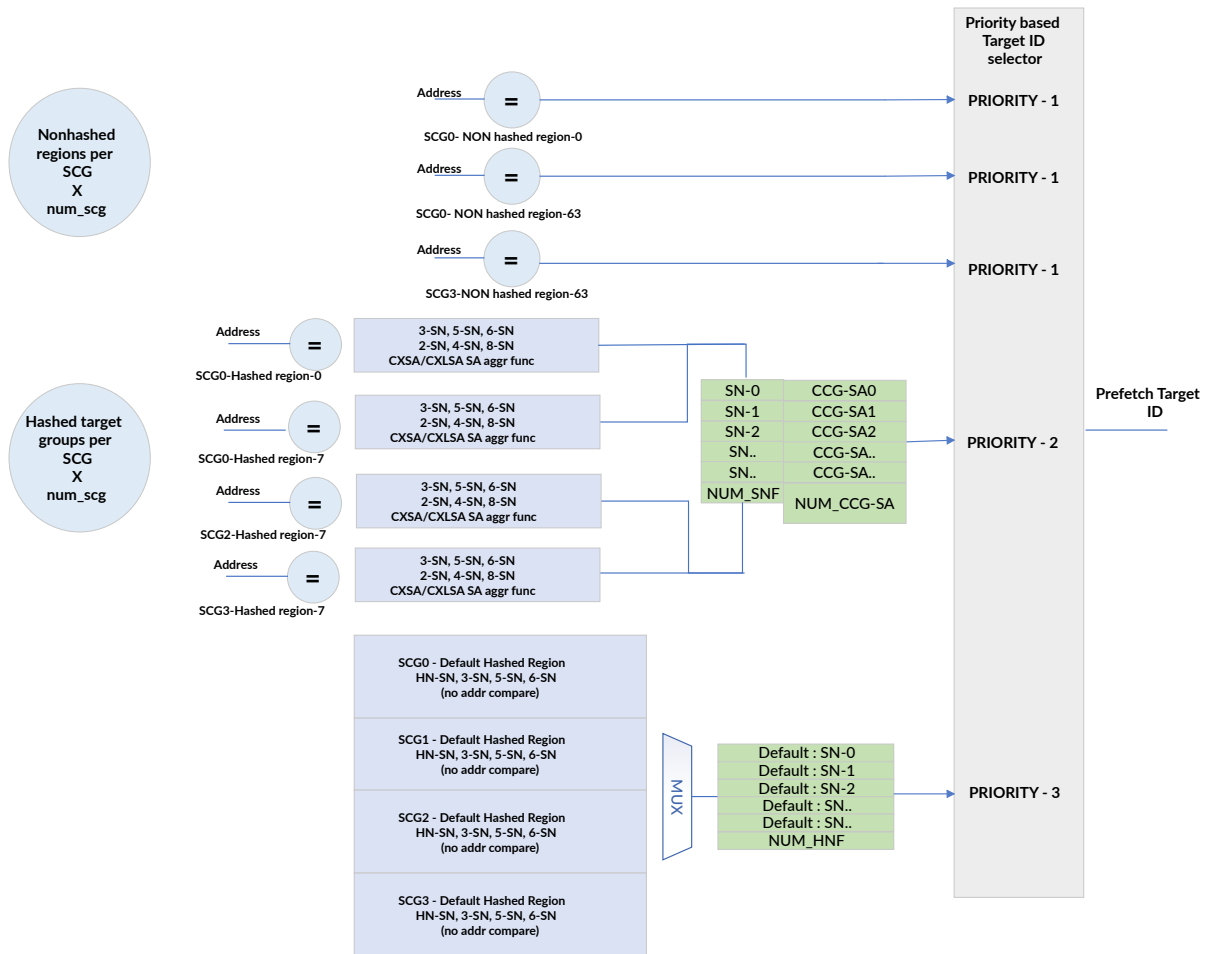
Figure 3-7: Example of HN-P flexible target ID configuration

HTG3-HN-P base index=0 HTG3 : num_HN-P configured = 8	HN-P Hashed Tgt-ID table
	Hashed Tgt-ID (0)
	Hashed Tgt-ID (1)
	Hashed Tgt-ID (2)
	Hashed Tgt-ID (...)
HTG4-HN-P base index=8	Hashed Tgt-ID (8)
	Hashed Tgt-ID (...)
	Hashed Tgt-ID (NUM_HNP)

3.4.2.2 Support for PrefetchTarget operations in RN SAM

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

Figure 3-8: RNSAM Prefetch target ID



To support PrefetchTarget operations, the RN SAM integrates the functionality of HN-F SAM to determine the appropriate SN-F target ID for a given address. RN SAMs that integrate PrefetchTarget support only use the SN-F target ID for PrefetchTarget requests. The PrefetchTarget RN SAM programming must match the HN-F SAM for SN-F target IDs. PrefetchTarget support in the RN SAMs is enabled by default and supports PrefetchTarget IDs for a maximum of 8 SCGs, determined by the `RNSAM_PFTGT_NUM_SCG` parameter.

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

- Direct-mapped or address-striped in the HN-F SAM (Default Hashed Region)

- Non-hashed target regions in the HN-F
- HTG region in the HN-F

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

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

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

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

Support for PrefetchTarget operations to address range-based targets

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

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

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

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



Note

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

Table 3-6: Mapping of SCGs to QoS region registers

SCG ID	Mapped registers
SCG0	<ul style="list-style-type: none"> • <code>sam_qos_mem_region_reg0</code> • <code>sam_qos_mem_region_reg1</code>
SCG1	<ul style="list-style-type: none"> • <code>sam_qos_mem_region_reg2</code> • <code>sam_qos_mem_region_reg3</code>
SCG2	<ul style="list-style-type: none"> • <code>sam_qos_mem_region_reg4</code> • <code>sam_qos_mem_region_reg5</code>
SCG3	<ul style="list-style-type: none"> • <code>sam_qos_mem_region_reg6</code> • <code>sam_qos_mem_region_reg7</code>

PrefetchTarget operations based on HN-F Non-hashed regions

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

$$\text{RNSAM_PFTGT_NUM_SCG} \times \text{RNSAM_PFTGT_NUM_NONHASH_PSCG}$$

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

Prefetch non-hashed regions can be programmed as a standalone QOS override regions and not determine prefetch targets. This is achieved by disabling the prefetch enable bit for each prefetch non-hashed region. Prefetch non-hashed region by default enable prefetch functionality and disables QOS override functionality.

PrefetchTarget operations based on HN-F HTG regions

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

$$\text{RNSAM_PFTGT_NUM_SCG} \times \text{RNSAM_PFTGT_NUM_HTG_PSCG}$$

Prefetched hashed and non-hashed memory regions are expected not to have any overlap space among them.

Prefetch Hashed region supports the following hash functions: MOD-3/5/6 hashing, Power of two hashing (2SN, 4SN, 8SN). For more information about these hashing mechanisms see [3.4.3 HN-F SAM](#) on page 54.

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

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

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

SN-TargetID lookup for the PrefetchTarget operations

PrefetchTarget operations targeting hashed regions and default hashed derive the SN target ID from the configured SN Target ID tables.

HTG regions

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

Default hashed region

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

3.4.2.3 HN-F with CAL support

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

The HN-F NodeIDs paired at the CAL are only differentiated using the device ID (NodeID[1:0]) field in the node ID

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



Note

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

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

Normal mode

In normal mode, each HN-F node ID is explicitly assigned to an SCG using the methods that [3.4.2.1 Hashed Target Groups](#) on page 29 describes. In the example configuration, all eight HN-F node IDs are entered in the target ID registers of the SCG.

The HN-F count field for that SCG would be set to eight. This approach allows up to 128 HN-F nodes that are connected to 64 CAL instances to be assigned to an SCG.

CAL mode

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

Mixed mode

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

The following table lists the options.

Table 3-7: HN-F CAL/non-CAL options

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



Note

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

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

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

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

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

Number of bits in select: 3

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

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

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

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

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

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

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

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

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

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

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

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

The following limitations apply when using CAL mode:

- This feature must only be used when the target HN-F are paired using CAL.
- RN SAM does not apply this method to SN-F Target IDs. To fully utilize CHI PrefetchTarget operations to SN-F, the paired HN-Fs must always be mapped to same SN-F or group of SN-Fs if 3-SN, 5-SN, or 6-SN hashing is used.
- Only the lowest value HN-F ID from each CAL group must be programmed in the RN SAM hashed Target ID registers.

- If an SCG contains a mix of local HN-F and CCG NodeIDs, HN-F CAL mode must not be used for that SCG.

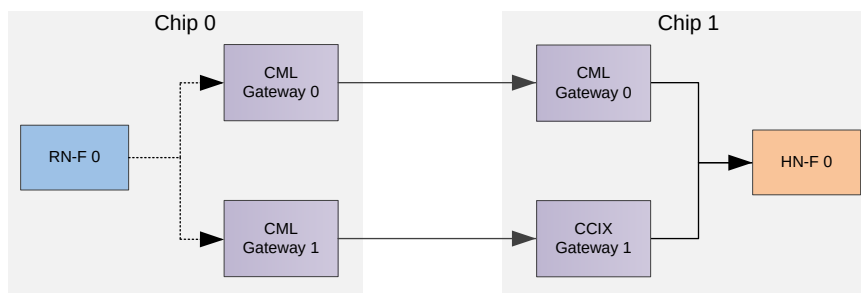
3.4.2.4 SAM support for CML Port Aggregation

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

Support for CPA in RN SAM

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

Figure 3-9: RN SAM CML Port Aggregation



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

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

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

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

CPAG supports following hashing ways:

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

- non-power of two (mod-3 way) hashing

Support for CPA in HN-F SAM

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

HN-F uses the *Logical ID* (LDID) of the RN-F to determine whether CPA is enabled. As [3.4.5 Cross chip routing and ID mapping](#) on page 79 describes, HN-F contains the LDID to physical node ID conversion table as [Table 3-21: Example program](#) on page 83 shows in the Example Programming table. This table, along with the CHI node ID and valid fields, also contains remote, cpa_en, and cpa_grpid bits. HN-F uses these bits to determine whether the RN-F is enabled to use CPA. It then sends the snoops through appropriate ports by hashing the address bits. To enable CPA for the ID of each RN-F, see [3.4.3 HN-F SAM](#) on page 54.

AXID based CML port aggregation

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

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

- CPA can only be used for HTG and non-hashed address ranges in the RN SAM
- CPA must not be enabled for Device nR memory traffic. CPA must not be enabled for an address range that can be changed from Normal to Device nR. This can happen with updates to page tables, where a given memory region is changed from Normal to Device nR.
- Each non-hashed memory range can be explicitly enabled to use CPA or use a single non-hashed target ID
- The target ID of each HTG can be explicitly enabled to use CPA
- The HN-F, when participating as a CPA target for traffic from a remote RN-F, must not receive non-CPA traffic from the same remote RN-F. This requirement means that an RN-F cannot send CPA and non-CPA traffic to the same remote HN-F.
- cml_port_aggr_grp_reg<X> registers contain the full list of CML gateway target IDs
- HTG in RN SAM supports multiple CPA groups. Each HN-F can be programmed to associate to different CPA group.
- CPA_EN for each HN-F is specified by programming bit vector to register sys_cache_grp_hn_cpa_en_reg (0-63 HNFs) and hashed_target_grp_hnf_cpa_en_reg1 (64-128 HNFs).
- Multiple CPA groups are enabled by programming each HTG sys_cache_grp_hn_cpa_grp_reg#. enable_multi_cpa_grp_scg# = 1'b1. CPA group per each HN-F is specified by programming group ID vector to hashed_target_grp_cpag_perhnf_reg#.
- If HTG is enabling only one CPA group (Legacy mode) then programming is simplified by disabling sys_cache_grp_hn_cpa_grp_reg. enable_multi_cpa_grp_scg# = 1'b0 and programming CPA group per each HTG sys_cache_grp_hn_cpa_grp_reg#.cpa_grp_scg#

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

AXID based hashing is not supported inside of the CPAG.

3.4.2.5 Address bit masking in the RN SAM

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

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

- `rnsam_hash_addr_mask_reg`
- `rnsam_region_cmp_addr_mask_reg`

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

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

By programming select bits to 0b0 in the `rnsam_region_cmp_addr_mask_reg` mask register, both the incoming address and the programmed address ranges can be masked off before comparison. This region mask is applied to hashed, non-hashed, and GIC memory regions.

RN SAM hashes all address bits [MSB:6] to equally distribute the requests across all HN-F and SN-F target devices. By programming select bits in the `rnsam_hash_addr_mask_reg` address mask register to 0b0, those address bits can be removed from the hashing logic. This feature is only applicable to hashed memory regions.

The following limitations apply:

- 3-SN, 5-SN, and 6-SN mode is enabled, a configurable group of nine address bits in the range of address bits [21:8] and the configured `top_addr_bits` are essential in distributing the addresses between memory. Arm recommends that these bits are masked carefully to avoid memory aliasing. See [3.4.3.1 HN-F to SN-F memory striping in HN-F SAM](#) on page 59
- Range compare mask must not mask off bits that represent the size of the region. For example, if any of the region sizes are 64MB, address bit 26 cannot be masked. Similarly, if a region size is 512MB, address bit 29 cannot be masked.
- The address bits masked in HN-F and RNSAM must be consistent. This requirement ensures that PrefetchTarget requests from an RN-F to SN-F are addresses by the same SN-F as SLC miss from the HN-F to SN-F.

Table 3-8: RN SAM parameters and configuration registers

RN SAM feature	Applicable parameters	Applicable configuration registers
Global	RNSAM_FLEX_TGTID_EN	rnsam_status rnsam_region_cmp_addr_mask_reg
GIC region	-	gic_mem_region_reg
Non-hashed regions	RNSAM_NUM_NONHASH_REGION	non_hash_mem_region_reg0-63
	RNSAM_NONHASH_RCOMP_EN	non_hash_mem_region_cfg2_reg0-63
	RNSAM_NONHASH_RCOMP_LSB	non_hash_tgt_nodeid0-15
Hashed regions (HN-F)	RNSAM_NUM_HTG	rnsam_hash_addr_mask_reg
	RNSAM_HIER_HASH_EN	sys_cache_grp_region0-3
	RNSAM_HTG_RCOMP_EN	hashed_tgt_grp_cfg1_region4-31
	RNSAM_HTG_RCOMP_LSB	hashed_tgt_grp_cfg2_region0-31
	RNSAM_NUM_ADD_HASHED_TGT	hashed_tgt_grp_cfg2_region0-31 sys_cache_grp_secondary_reg0-3 hashed_target_grp_secondary_cfg1_reg4-31 hashed_target_grp_secondary_cfg2_reg0-31 sys_cache_group_hn_count hashed_target_group_hn_count_reg1-3 sys_cache_grp_nonhash_nodeid hashed_target_grp_nonhash_nodeid_reg1-6 sys_cache_grp_hn_nodeid_reg0-15 hashed_target_grp_hnf_nodeid_reg16-31 sys_cache_grp_cal_mode_reg hashed_target_grp_cal_mode_reg1-7 hashed_target_grp_hash_cntl_reg0-31 hashed_target_grp_hnf_device_bound_cfg_reg0-1
Hashed regions (HN-P/CCG)	RNSAM_AXID_HASH_EN	rnsam_hash_axi_id_mask_reg hashed_target_grp_hnp_nodeid_reg0-15 hashed_target_grp_hnp_cpa_en_reg0
QoS regions	RNSAM_NUM_QOS_REGIONS	sam_qos_mem_region_reg0-7
		sam_qos_mem_region_cfg2_reg0-7

RN SAM feature	Applicable parameters	Applicable configuration registers
CPAG support	RNSAM_NUM_CPA_GRP	sys_cache_grp_hn_cpa_en_reg hashed_target_grp_hnf_cpa_en_reg1 sys_cache_grp_hn_cpa_grp_reg hashed_target_grp_cpa_grp_reg1-7 cml_port_aggr_mode_ctrl_reg cml_port_aggr_mode_ctrl_reg1-6 cml_port_aggr_grp0-31_add_mask cml_port_aggr_grp_reg0-12 cml_port_aggr_ctrl_reg cml_port_aggr_ctrl_reg1-6
PrefetchTgt support	RNSAM_PREFETCH_EN	sys_cache_grp_sn_nodeid_reg0-31
	RNSAM_PFTGT_NUM_SCG	sys_cache_grp_hashed_regions_sn_nodeid_reg0-15
	RNSAM_PFTGT_NUM_NONHASH_PSCG	sys_cache_grp_sn_attr
	RNSAM_PFTGT_NUM_HTG_PSCG	sys_cache_grp_sn_attr1
	RNSAM_PFTGT_DEF_HASHED_GRP_EN	sys_cache_grp_sn_sam_cfg0-3 sys_cache_grp_region0_sn_nodeid_reg0-31 sys_cache_grp_region1_sn_nodeid_reg0-31 sam_scg0-7_prefetch_nonhashed_mem_region_cfg1_reg0-63 sam_scg0-7_prefetch_nonhashed_mem_region_cfg2_reg0-63 sam_scg0-7_prefetch_hashed_region_cfg1_reg0-7 sam_scg0-7_prefetch_hashed_region_cfg2_reg0-7 sam_scg0-7_prefetch_hashed_region_cfg3_reg0-7
Custom hash	RNSAM_CUSTOM_REG	sam_generic_regs0-7



- Parameters and registers in the same row are not necessarily related
- Ranges in the register names, m-n, indicates the register repeats (m-n+1) times
- Register names with two ranges, k-1 and m-n, indicates the register repeats $(l - k + 1) \times (m - n + 1)$ times

3.4.2.6 RN SAM Compact HN Tables for multi-chip configurations

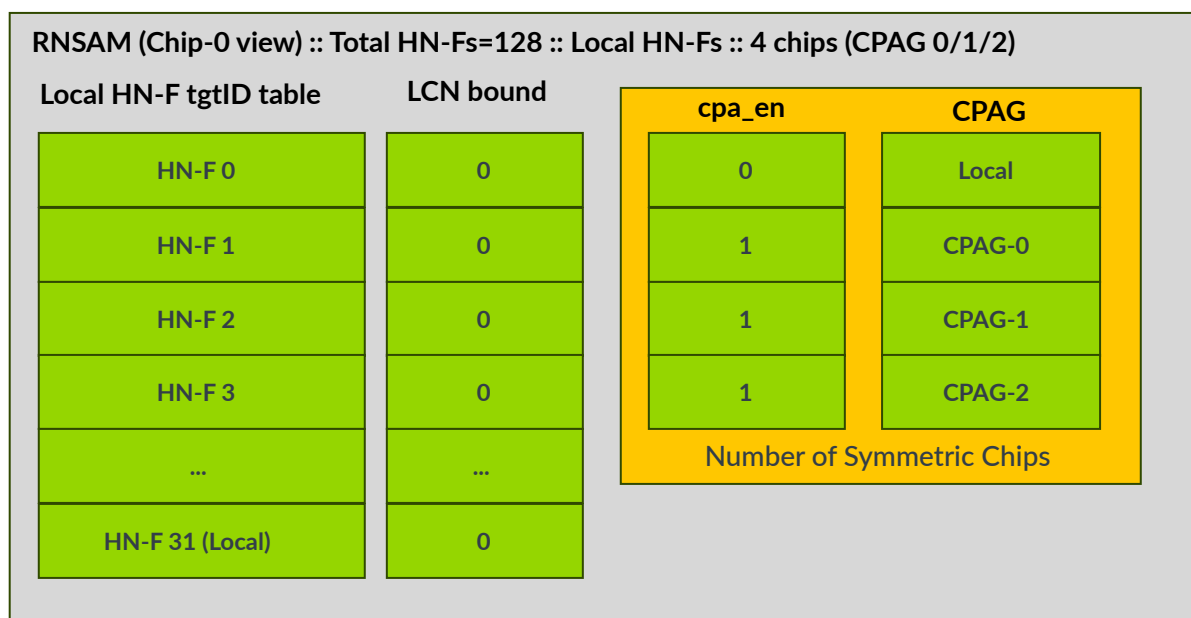
The CMN-700 multi-chip configurations with single NUMA, hashing across local and remote HN-Fs RN SAM logic requires only local HN-F target-ids to be programmed. The RN SAM targets CCG node-ids for the remote HN-F target-ids.

Considering only local HN-F target-ids you can build the SAM with minimal flops.

This mode is enabled by setting the parameter `COMPACT_HN_TABLES_EN_PARAM`. The Compact HN tables mode is contrary to the legacy modes where SAM build flops for both the local and remote HN-Fs. Compact HN tables mode is limited only multi-chip system with the same HN counts across all the chips. The local HN-F target-id table look up bits are derived from the total HN-F hash selection bits. The following figure shows an example of the Compact HN table based on a configuration with 128 HNFs and 4 symmetric chips.

For Compact HN tables mode, only power-of-two and Hierarchical hashing modes are supported.

Figure 3-10: Compact HN tables



The local HN-F and CPA Group hash index is determined by shuttering the bits from total HN-F hash index bits based on the configuration. for example The SAM performs a hash for the entire 128 HNFs = 7bit hash index bits [6:0]:

- CPAG hash index = total HN-F hash index [1:0] = 4 symmetrical chips = 2 bits
- Local HN-F hash index = total HN-F hash index [6:2] = 32 local HN-Fs = 5 bits

For Hierarchical hashing, the shuttering of hash bits only applies to the cluster hash (first hash). for example The SAM performs hierarchical hashing across 128 HN-Fs (16 clusters x 8 HN-Fs per cluster)

The cluster hash index (first hash) = [3:0] (16 cluster)

HN-F nodes hashes the index within a cluster = [2:0] (8 nodes per cluster)

- CPAG hash index = cluster hash index [1:0] = 4 symmetrical chips = 2 bits
- Local HN-F hash index = cluster hash index [3:2] = 4 cluster per chip = 32 local HN-F



All the HN-Fs within a cluster should be within one chip.

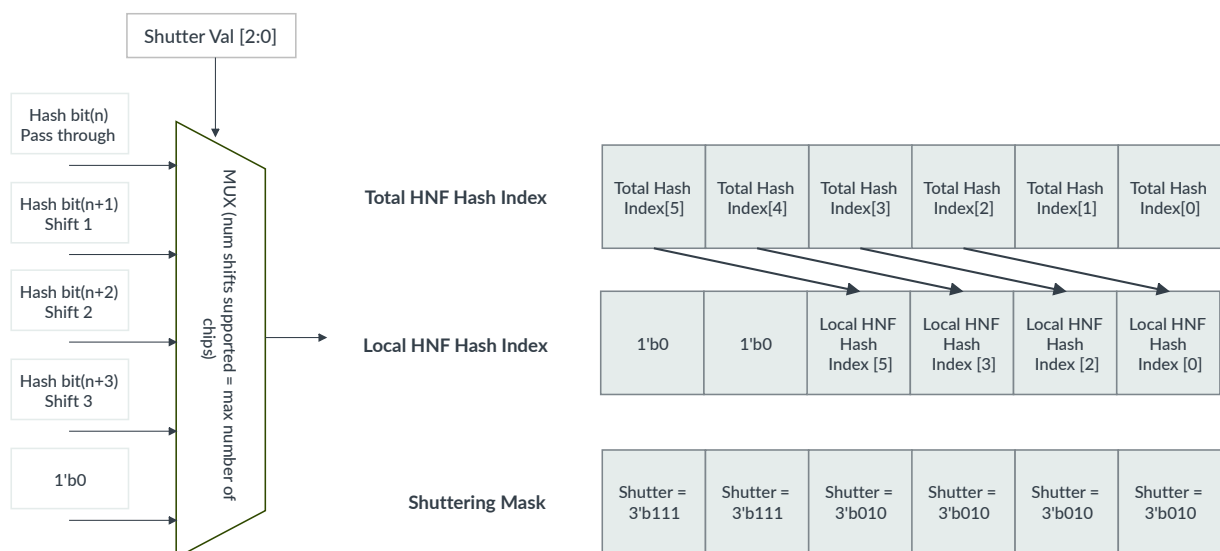
3.4.2.6.1 Local HN-F and CPAG hash index: shuttering from total hash index bits

To configure shuttering value for each Local HN-F hash index and CPAG hash index.

- Total HN-F hash index bits: Supports maximum of 128 HN-Fs (256 HN-F in CAL2) - 7 bits
- Total HN-F hash index bits: Supports maximum of 64 HN-Fs per chip (128 HN-F in CAL2) - 6 bits
- CPAG hash index bits: Supports maximum of 32 symmetrical chips – 5bits

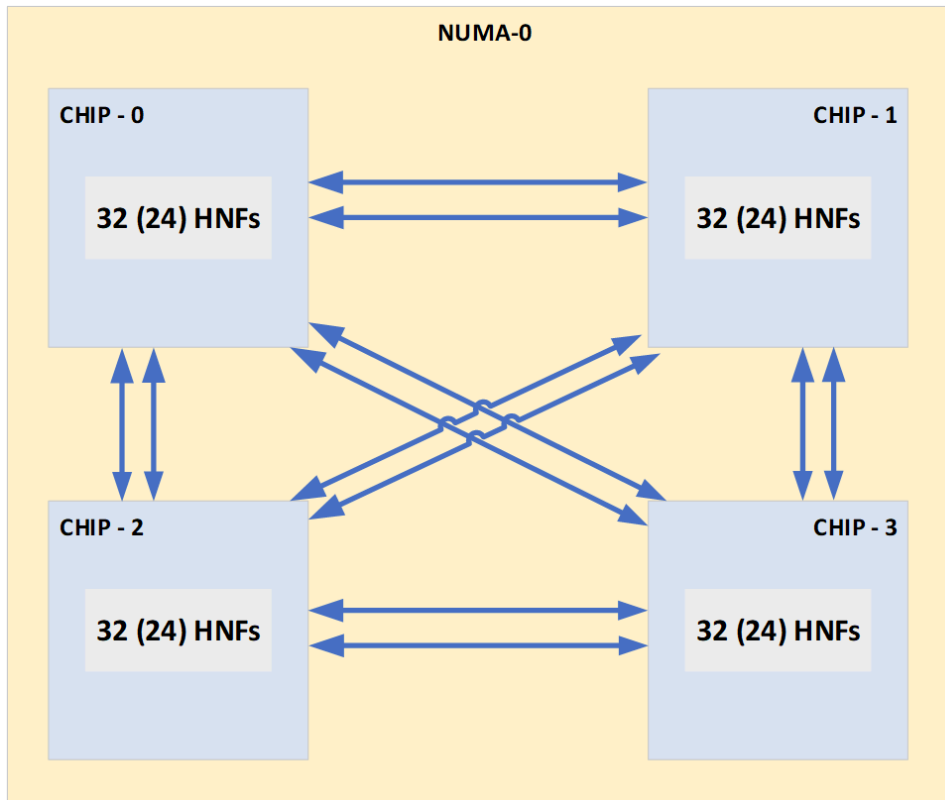
The SAM uses programming structures to determine HN-F selection, where hash bits are selectively picked from the overall hash bit, as shows the following figure:

Figure 3-11: SAM programming structures to choose the HN-F selection bits from overall hash bits



The following figure shows a programming of a four Chip SMP in a single NUMA configuration, which includes a power of two HN-Fs and non-power of two HN-Fs, using 24 as an example.

Figure 3-12: 4 CHIP SMP (1 NUMA)



The following table shows the programmed configuration fields for a power of two HN-F Hashed Target Group for the previous single NUMA figure.

Table 3-9: Single Numa power of two Hashed Target Group configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	128	hnf_hash[6:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hns_sel_shuttering[6:0][3:0]	{value}	hnf_hash[6:2]
cpa_grpid_shuttering[4:0][3:0]	{value}	hnf_hash[1:0]
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hns_base_index	0	-

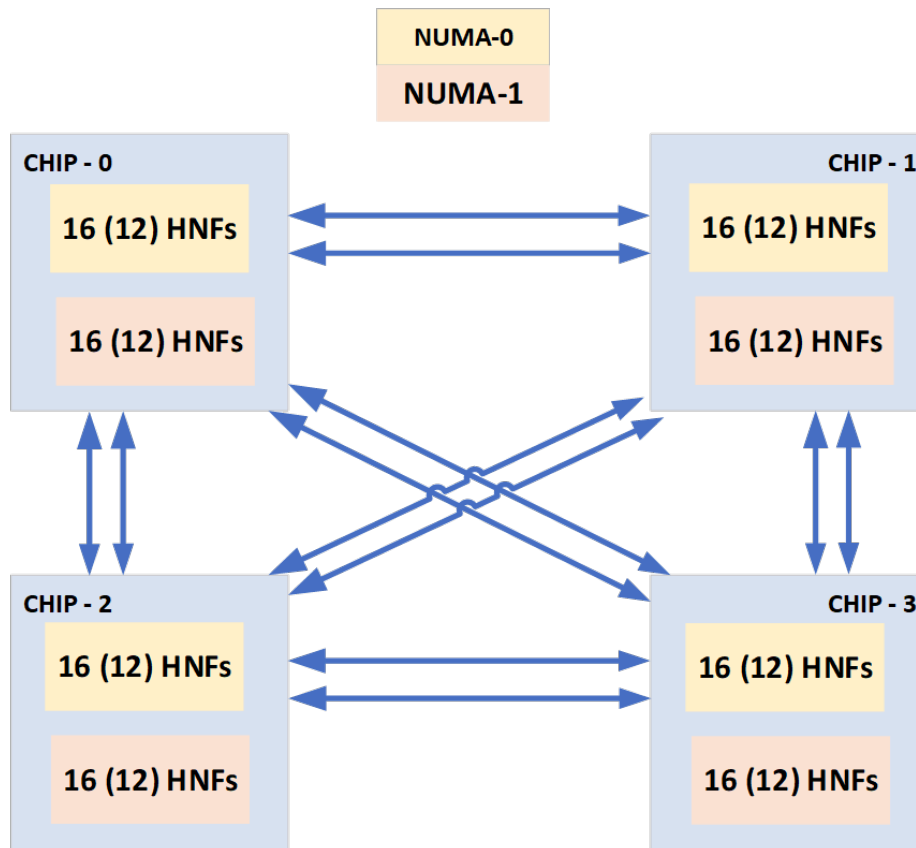
The following table shows the programmed configuration fields for a non-power of two HN-F Hashed Target Group for the previous single NUMA figure.

Table 3-10: Single Numa non-power of two Hashed Target Group configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	96	-
hierarchical_hash_en	1	-
hier_hash_cluster	8	clstr_hash[2:0]
hier_hash_nodes	12	node_hash[3:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hier_addr_shuttering[51:6][2:0]	{value}	addr[51:9]
hns_sel_shuttering[6:0][3:0]	{value}	clstr_hash[2] x node_hash[3:0]
cpa_grpid_shuttering[4:0][3:0]	{value}	hnf_hash[1:0]
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hns_base_index	0	-

The following figure shows a programming of a Multi Chip SMP in a two NUMA configuration, which includes a power of two HN-Fs and non-power of two HN-Fs using 12 as an example.

Figure 3-13: Multi CHIP SMP (2 NUMA)



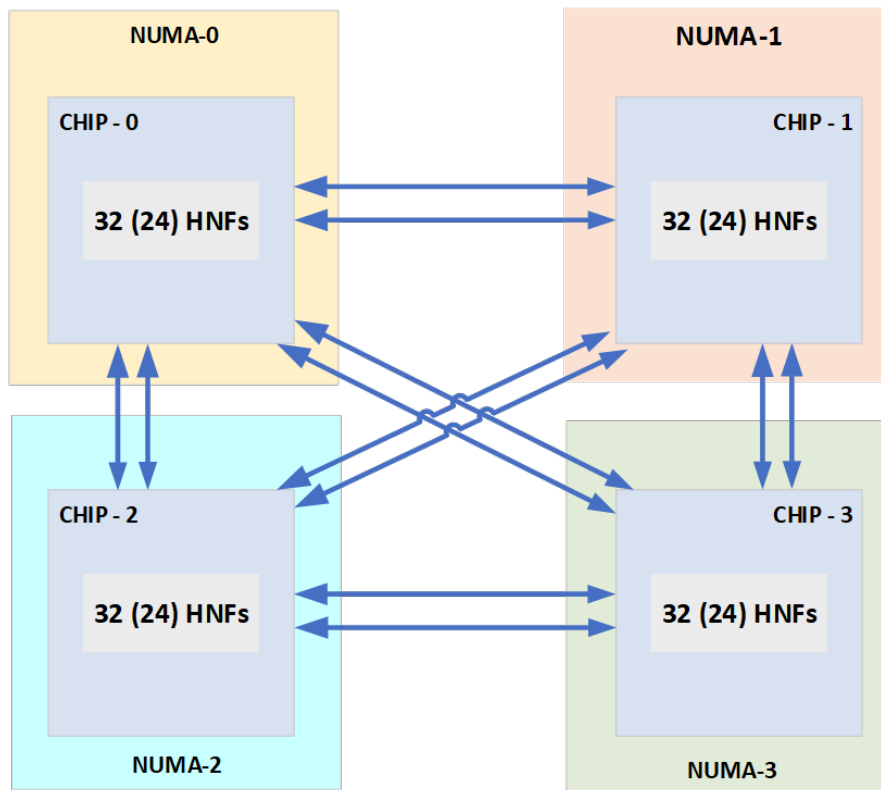
The following table shows the programmed configuration fields for a power of two HN-F Hashed Target Group for the previous two NUMA figure.

Table 3-11: Two Numa power of two Hashed Target Group[0/1] configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	64	hnf_hash[5:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hns_sel_shuttering[6:0][3:0]	{value}	hnf_hash[5:2]
cpa_grpid_shuttering[4:0][3:0]	{value}	hnf_hash[1:0]
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hns_base_index	0	-

The following figure shows a programming of a four Chip SMP in a four NUMA configuration, which includes a power of two HN-Fs and non-power of two HN-Fs using 24 as an example.

Figure 3-14: 4 CHIP SMP (4 NUMA)



The following table shows the programmed configuration fields for a power of two HN-F Hashed Target Group for the previous four NUMA figure.

Table 3-12: Two Numa power of two Hashed Target Group[0/1/2/3] configuration fields

Configuration Field	Values	Description
target_type	HN-F	-
num_hnf	64	hnf_hash[5:0]
htg_hnf_nodeids[31:0][10:0]	{value}	HN-F target ID
hns_sel_shuttering[6:0][3:0]	{value}	hnf_hash[4:0]
cpa_grpid_shuttering[4:0][3:0]	{value}	always point to index - 0
cpa_en[15:0]	{1, 1, 1, 0}	CPA_EN table
cpa_grpid[15:0][4:0]	{CPAG2, CPAG1, CPAG0, -}	CPA grpid table
hns_base_index	0	-

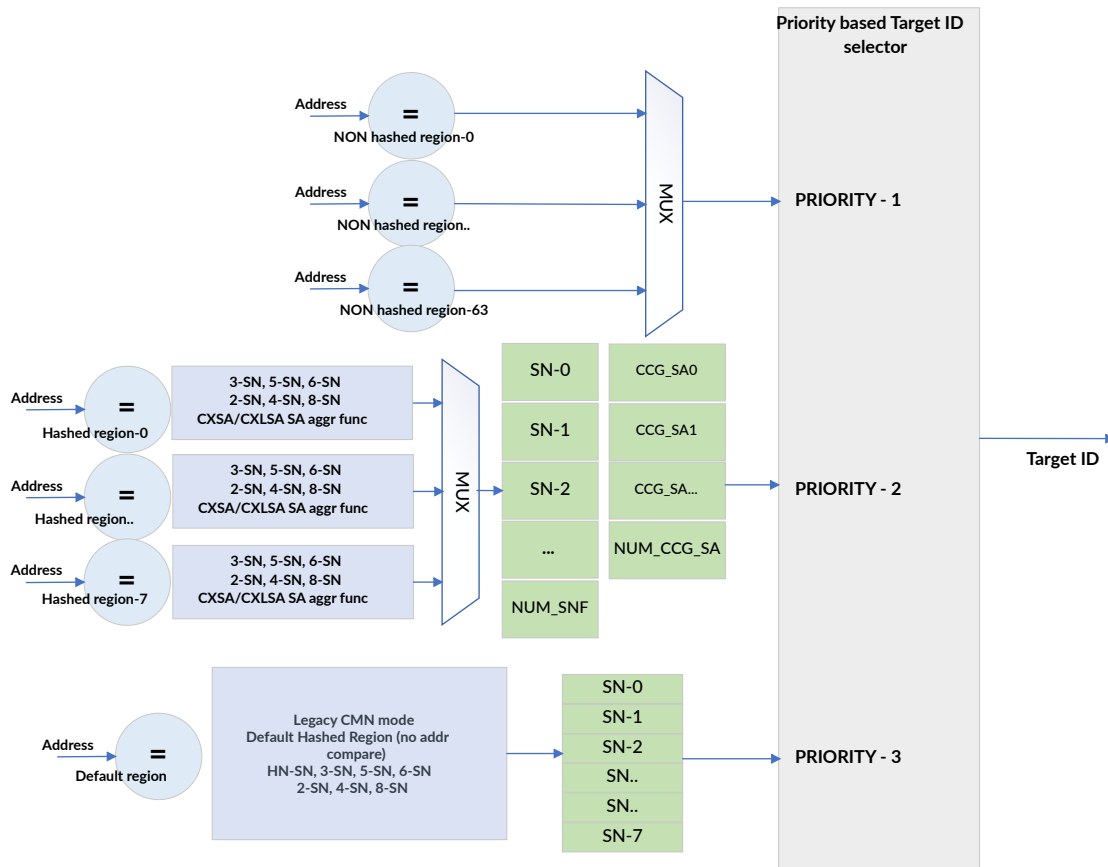
For more information about programming the RNSAM for 2 NUMA and 4 NUMA configurations for HTG and CPA functionality, see [3.8.1.1 RNSAM programming for 2P \(2 NUMA\) HTG and CPA functionality for HNS systems](#): on page 103 and [3.8.1.2 RNSAM programming for 4P \(4 NUMA\) HTG and CPA functionality for HNS systems](#): on page 105

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

Figure 3-15: HN-F SAM target ID selection policy



The preceding figure shows that the HN-F SAM has the following mapping policies to generate target IDs for transactions:

- Range-based:
 - Non-hashed
 - Hashed target groups
- Default hashed region (Legacy CMN mode):
 - Hashed
 - Direct mapping

An HN-F cannot use both hashed mapping mode and direct mapping mode. To map transactions that fall in the range-based part of the HN-F SAM, HN-F SAMs support priority-based target ID selection. The order of priority when selecting a target ID is:

1. Range-based: non-hashed SN target ID
2. Range-based: hashed SN target ID
3. Default Hashed region: Hashed target ID or direct mapped target ID (Legacy CMN mode).

HNSAM range-based regions support two ways of defining a memory region and it is selected through a user parameter `HNSAM_RCOMP_EN`.

- Base address & region size, size of the region being power of two (64MB minimum region size; `HNSAM_RCOMP_EN = 0`).
- Lower address & upper address, no restrictions on size of the region. (`HNSAM_RCOMP_EN = 1`)

Minimum size of the memory region is selected through a user parameter (`HNSAM_RCOMP_LSB = [20-26]`) that defines the lower address bit for the comparison. This is limited to the lower address & upper address mode only.

- `HNSAM_RCOMP_LSB = 20`, defines minimum memory size = 1MB
- `HNSAM_RCOMP_LSB = 21`, defines minimum memory size = 2MB
- `HNSAM_RCOMP_LSB = 22`, defines minimum memory size = 4MB
- `HNSAM_RCOMP_LSB = 26`, defines minimum memory size = 64MB

Range-based mapping: Non-hashed target IDs

Non-hashed range-based mapping is an address-based unique target ID generation policy. Up to 64 memory regions (`HNSAM_NUM_NONHASH <= 64`) can be created, each targeting a single SN. This mode is useful where a partition of memory from the global DRAM is mapped explicitly to an individual SN, for example, an on-chip SRAM.

Range-based mapping: Hashed target groups

Hashed range-based mapping is an address based hashed target ID generation policy. Up to eight memory regions (`HNSAM_NUM_HTG <= 8`) can be created, each targeting a group of SNs. This mode is useful when a partition of memory from the global DRAM space is mapped explicitly to a group of SNs. These modes can only be used when a DRAM partition targets the following SN configurations:

- 3-SN mode: Addresses from a hashed region are striped across three SNs
- 5-SN mode: Addresses from a hashed region are striped across five SNs
- 6-SN mode: Addresses from a hashed region are striped across six SNs
- 2-SN mode: Addresses from a hashed region are striped across two SNs
- 4-SN mode: Addresses from a hashed region are striped across four SNs
- 8-SN mode: Addresses from a hashed region are striped across eight SNs

In 3-SN, 5-SN, or 6-SN mode, addresses are striped at a 256B granularity between the 3, 5, and 6 SNs. The stripe function uses a configurable group of nine address bits in the range of address bits [21:8] and an extra two (3-SN) or three (5-SN, 6-SN) user-defined address bits.

In 2-SN, 4-SN, or 8-SN mode, addresses are striped at a 64B granularity between the 2SNs/4SNs/8SNs. The stripe function uses XOR of address bits [MSB:6].

- Two SNs:
 - Number of bits in select: 1
 - $\text{Select}[0] = (6^7 7^8 \dots^{51})$

- Four SNs:
 - Number of bits in select: 2
 - Select [0] = $(6^8 10^{\dots 50})$
 - Select [1] = $(7^9 11^{\dots 51})$
- Eight SNs:
 - Number of bits in select: 3
 - Select [0] = $(6^9 12^{\dots 51})$
 - Select [1] = $(7^{10} 13^{\dots 49})$
 - Select [2] = $(8^{11} 14^{\dots 50})$

Default Hashed region: Hashed mapping (legacy CMN mode)

The HN-F SAM Legacy mode (`HNSAM_DEF_HASHED_GRP_EN = 1`) supports the following hashed modes for SN target ID selection. The default hashed region does not use address comparison. When an incoming address does not match any range-based mapping, then SN TargetID is derived from the default hashed region:

2-SN mode

Addresses from a given HN-F are striped across two SNs

3-SN mode

Addresses from a given HN-F are striped across three SNs

4-SN mode

Addresses from a given HN-F are striped across four SNs

5-SN mode

Addresses from a given HN-F are striped across five SNs

6-SN mode

Addresses from a given HN-F are striped across six SNs

8-SN mode

Addresses from a given HN-F are striped across eight SNs

The default hashed region has address comparison logic added to detect for incorrect address region access. The default hashed regions address boundaries must match the RNSAM HTG address boundaries.

The HNSAM notifies the HN-F upon an address compare miss if the address does not match any programmed SAM ranges.

The HN-F does not generate NDE (non-data error) responses on decode error.

- Write response: The HN-F will drop the response
- Read response: The HN-F sends the response back to the RN-F or CCG with poison.

To maintain backwards compatibility, program the `base_addr = 0x0` and `size = 0xFF` or `end_addr = 0xFFFFFFFF`, depending on which RCOMP parameter is used.

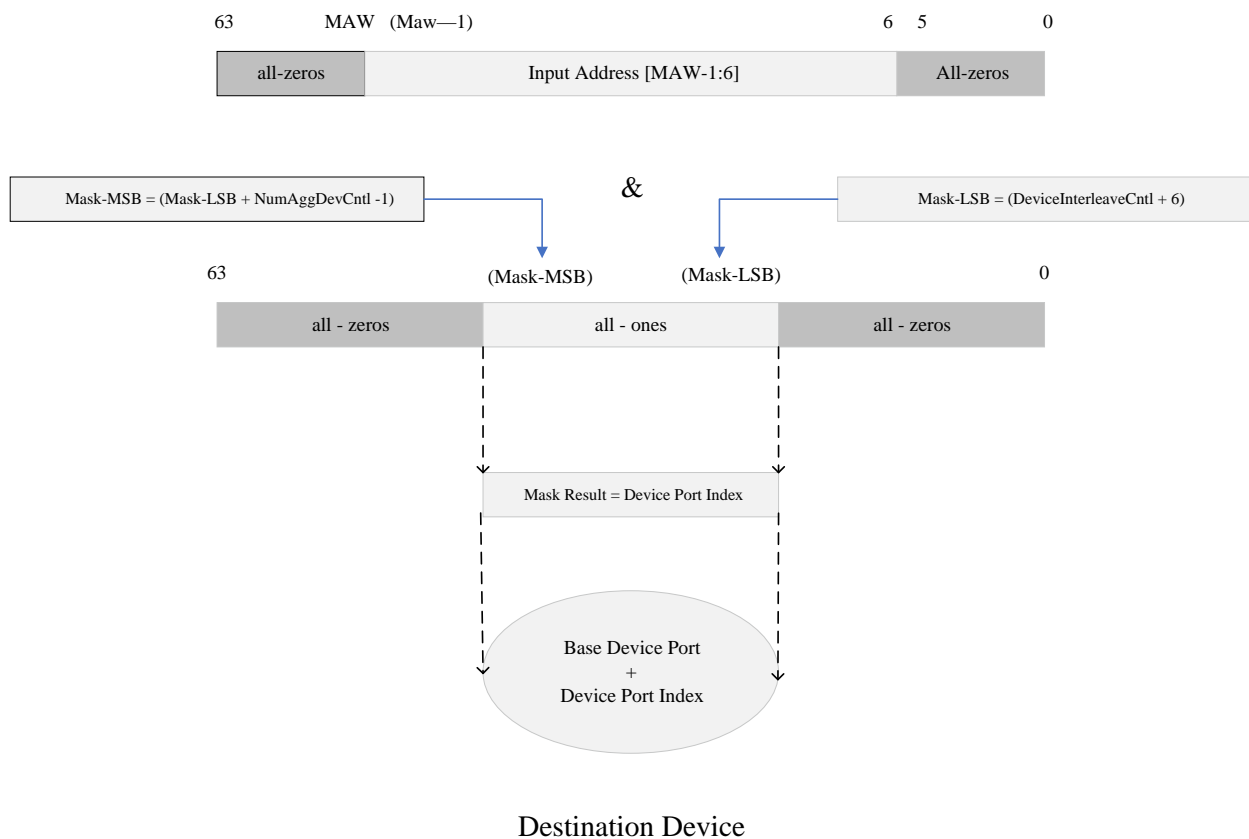
Direct mapping

Direct SN mapping is used if the SCG targets 1, 2, 4, 8, 16, 32 SNs. When attached using CAL, 64 SNs are supported. In this case, the transaction uses the SNO target ID. Distributing accesses across the SNs targeted by the SCG is achieved by programming the SNO field of each HN-F to the SN node ID it targets. For example, if an SCG with eight HN-Fs targets eight SNs, the SNO field of each HN-F would be programmed with a different SN node ID. If that same SCG with eight HN-Fs targets four SNs instead, every two HN-F nodes would have the same SNO field value.

Aggregated SA selection function

HNSAM supports hashing across multiple CXL.MEM devices.

Figure 3-16: Aggregated SA selection function



Aggregated SA selection requires the following configurations to determine the SN target ID:

- Device interleaving: This configuration controls the interleave size across all aggregated CXSA devices and supports interleave sizes between 64 Bytes to 2Mbytes. This controls the number of LSB bits to be stripped from the address.
- Aggregated devices count: This configuration controls the number of the CXSA devices aggregated. The number of aggregated devices supported are 1, 2, 4, 8, 16, 3, 6, and 12 (power of two and modulus hashing modes are implemented as specified in the [Compute Express Link \(CXL\) Specification](#)).

- HNSAM supports maximum of 16 CXSA devices in the HN SAM HTG regions (CXSA port aggregation). Additional CXSA devices must be supported through HN SAM non-hashed regions.

Address bits extracted by stripping the LSB and MSB bits based on the previous configuration yields the selection bits for deriving the CXSA target ID.

User defined hashing logic

The hash modules can be selectively replaced to implement user-defined hash logic. HN SAM implements user-defined configuration registers to be used for the user-defined hash logic. The number of user-defined registers are enabled using a user parameter (`HNSAM_CUSTOM_REGS [0-2]`).

For more information about modifying the hash modules, see *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*.

SN Target-ID tables

HNSAM hashed target groups use the Target-IDs from the Target-ID structure with depth determined by number of SNs in the configuration. Maximum SN Target ID entries supported are 64.

HNSAM hashed target groups use CXSA Target IDs from the separate Target ID structure with depth determined by number of remote SNs in the configuration. The maximum CCG SA target ID entries supported are 16.

Each hashed target group generates the index based on the address striping and lookup into the target ID tables for the SN and CCG target ID. Base index for each HTG is derived based on the linked list mechanism from the boot time programming.

Hashed region[n] base index = (Hashed region [n-1] base index) + (Hashed region[n-1] num SN)

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

The CMN-700 HN-F SAM supports three non-power-of-two memory striping modes, which are known as 3-SN mode, 5-SN mode, and 6-SN mode respectively. In these modes, the HN-F stripes addresses across three SN-Fs, five SN-Fs, or six SN-Fs respectively.

All three striping modes use a configurable group of nine address bits in the range of address bits [21:8] and extra upper address bits configured as `top_addr_bits` as inputs to their stripe function. The address bit group used can be one of: [16:8] (Default), [17:9], [18:10], [19:11], [20:12], or [21:13].

3-SN mode

In 3-SN mode, a stripe function ensures that traffic is distributed evenly among the three SNs. The two higher PA bits are referred to as `top_address_bit1` and `top_address_bit0`. The top address bits are selected so that three of the four combinations of the top address bits appear evenly in the selected address space, and the fourth combination never appears.



In some situations, a top bit can be the inverse of the selected PA bit.

For each physical address, one of the three SNs is selected using the following formula when address bits [16:8] are used (default):

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

General SN distribution behavior example

For a simple case with a 3GB flat address space starting at address 0x0, top_address_bit1 is PA[31], and top_address_bit0 is PA[30]. With increasing physical address, the function steps between SNs at a 256-byte granularity. As the physical address iterates from 0-128KB, with top_address_bit1 = top_address_bit0 = 0, the first three terms distribute the traffic relatively evenly among the SNs. Of the 512 blocks, 256B each, in the first 128KB, the distribution is:

SN[0]	170 blocks 33.2%
SN[1]	171 blocks 33.4%
SN[2]	171 blocks 33.4%

This pattern repeats over each 128KB until 1GB, where top_address_bit0 toggles. With top_address_bit1 = 0 and top_address_bit0 = 1, the pattern is shifted. For each 128KB:

SN[0]	171 blocks 33.4%
SN[1]	170 blocks 33.2%
SN[2]	171 blocks 33.4%

At 2GB, when top_address_bit1 = 1 and top_address_bit0 = 0, the pattern shifts again:

SN[0]	171 blocks 33.4%
SN[1]	171 blocks 33.4%
SN[2]	170 blocks 33.2%

Over the full 3GB, the same number of lines are distributed to each SN.

The HN-F uses the hn_cfg_three_sn_en bit in its cmn_hns_sam_control register to enable routing to three SNs. In the cmn_hns_sam_control register, the hn_cfg_sam_top_address_bit0 and hn_cfg_sam_top_address_bit1 fields must be configured at boot time. These two address bits are decoded, and used with a hashing function to determine the target SN-F.

5-SN and 6-SN modes

Similar to 3-SN hashing, the 5-SN and 6-SN modes extend the function to equally distribute addresses between five or six SNs respectively. For each physical address, one of the three SNs is selected using the following formula when address bits [16:8] are used (default):

$$SN = \{ ADDR[10:8] + ADDR[13:11] + ADDR[16:14] + ((top_addr_bit2 < 2) | (top_addr_bit1 < 1) | top_addr_bit0) \} \% 6$$

HN-F SAM uses the following bits in the `cmn_hns_sam_control` register to enable striping across five or six SN-Fs:

- `hn_cfg_five_sn_en`
- `hn_cfg_six_sn_en`

In 5-SN and 6-SN mode, HN-F SAM also uses `hn_cfg_sam_top_address_bit2` field in the `cmn_hns_sam_control` register along with `hn_cfg_sam_top_address_bit1` and `hn_cfg_sam_top_address_bit0` to hash the incoming address.

3-SN, 5-SN, and 6-SN configurations



In each of the following examples, address bits [16:8] are used in the stripe function (the default).

This configuration ensures equal distribution of requests across all SN-Fs and prevents memory aliasing. The SAM also provides an `inv_top_address_bit` configuration bit, which can be used with top address bits. The following table shows the valid top address bits for Arm PDD Memory Map.

See the *Principles of Arm® Memory Maps White Paper*.

Table 3-13: 3-SN mode top address bits[bit 1, bit 0]

Combination 1 (<code>inv_top_address_bit</code> set to 0b0)	Combination 2 (<code>inv_top_address_bit</code> set to 0b0)	Combination 3 (<code>inv_top_address_bit</code> set to 0b1)	Combination 4 (<code>inv_top_address_bit</code> set to 0b1)
[0, 0]	[1, 1]	[0, 0]	[0, 0]
[0, 1]	[0, 1]	[1, 0]	[0, 1]
[1, 0]	[1, 0]	[1, 1]	[1, 1]



When `inv_top_address_bit=1`, it forces the SAM to invert the top most significant top address bit. For 3-SN mode, `top_address_bit1` is inverted. For 5-SN and 6-SN mode, `top_address_bit2` is inverted.

The following table shows the valid combinations for the address bits for 6-SN mode with PDD memory map.

Table 3-14: 6-SN mode top address bits[bit 2, bit 1, bit 0]

Combination 1 (<code>inv_top_address_bit</code> set to 0b0)	Combination 2 (<code>inv_top_address_bit</code> set to 0b0)	Combination 3 (<code>inv_top_address_bit</code> set to 0b1)
[0, 0, 0]	[0, 1, 0]	[0, 0, 0]
[0, 0, 1]	[0, 1, 1]	[0, 0, 1]

Combination 1 (inv_top_address_bit set to 0b0)	Combination 2 (inv_top_address_bit set to 0b0)	Combination 3 (inv_top_address_bit set to 0b1)
[0, 1, 0]	[1, 0, 0]	[0, 1, 0]
[0, 1, 1]	[1, 0, 1]	[0, 1, 1]
[1, 0, 0]	[1, 1, 0]	[1, 1, 0]
[1, 0, 1]	[1, 1, 1]	[1, 1, 1]

The combinations for 5-SN mode must follow similar rules to 6-SN mode programming. The top address bit combinations must be five sequential combinations from the preceding table.

Example 3-1: Example for PDD memory map

Assume that a system supports three SN-Fs with 32GB of DRAM at each SN-F port and all three SN-Fs are used for SCG 0. The first two regions together comprise a 32GB region, while the remaining two regions are 32GB each. Because the DRAM space is non-contiguous in this memory map (2GB + 30GB + 480GB), the base addresses for each DRAM partition are:

1. a. 000_8000_0000 to 000_FFFF_FFFF (2GB)
b. 008_8000_0000 to 00F_FFFF_FFFF (30GB)
2. 088_0000_0000 to 08F_FFFF_FFFF (32GB)
3. 090_0000_0000 to 097_FFFF_FFFF (32GB)

The following table breaks down the address bits for the regions that the previous list shows.

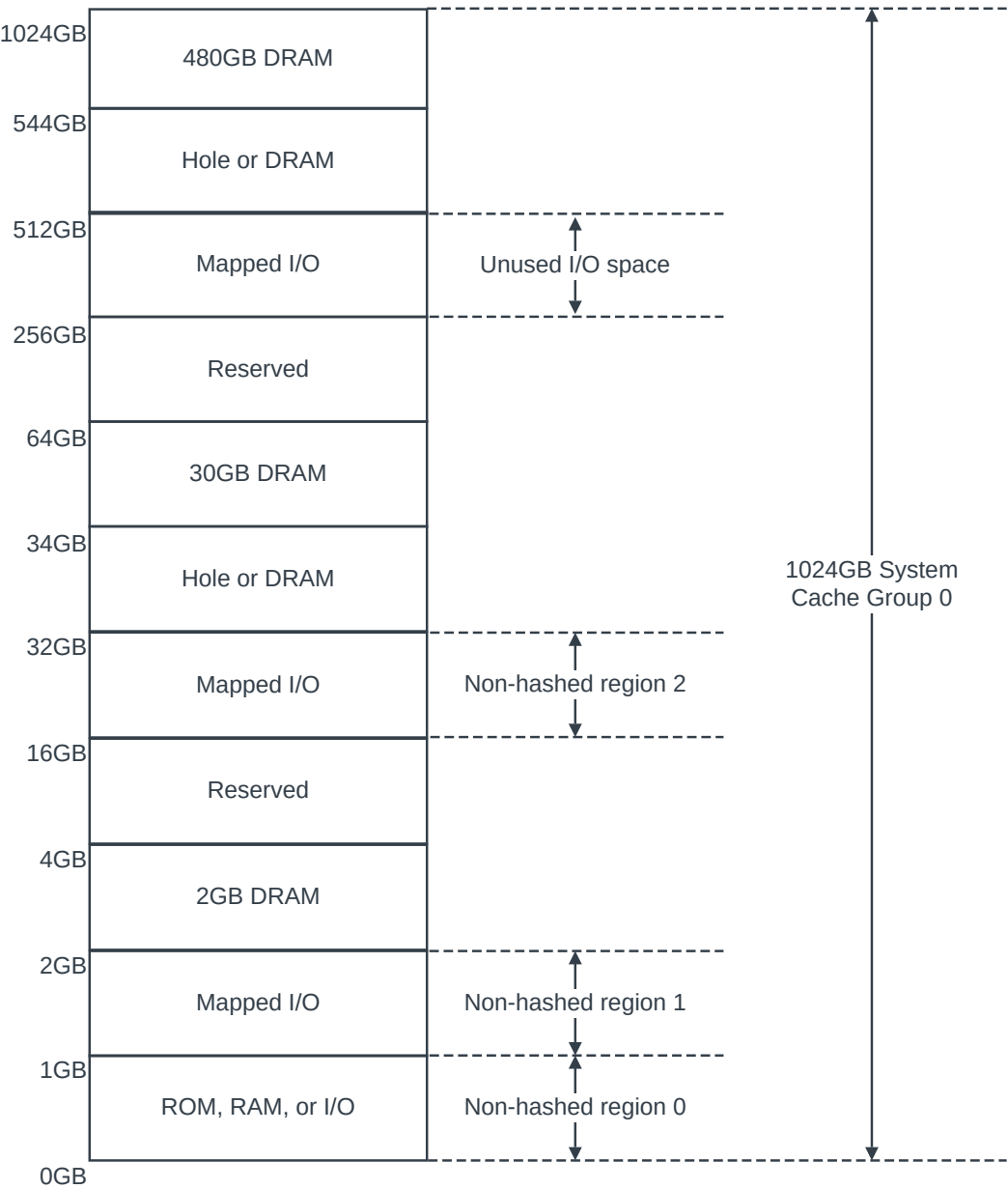
Table 3-15: Address region example bit settings: 3-SN example

Region	39	38	37	36	35	34	33	32	31
1a	0	0	0	0	0	0	0	0	1
1b	0	0	0	0	1	x	x	x	1
2	1	0	0	0	1	x	x	x	x
3	1	0	0	1	0	x	x	x	x

From the address bit breakdown, the selected top address bits must make sure both regions that are marked as Region 1 have the same values. This requirement ensures no aliasing in memory. For this memory map and DRAM size, there are no address bits that directly give Combination 1 or Combination 2 as shows [Table 3-13: 3-SN mode top address bits\[bit 1, bit 0\]](#) on page 61. However, if bits[39, 36] are used along with inv_top_address_bit = 1, then Combination 3 is possible. This approach ensures that the memory requests are equally distributed across the three SN-Fs without memory aliasing.

The following figure shows the Arm proposed memory map.

Figure 3-17: Example memory map programming



The following tables provide example address bits that provide equal distribution of memory across all SN-Fs in 3-SN and 6-SN hashed modes.

The following shows the 3-SN DRAM size settings.

Table 3-16: 3-SN DRAM size settings

3-SN DRAM size at each SN-F port	Top address bits[bit 1, bit 0]	Inv_top_address_bit value
1GB, total 3GB	[35, 30]	0b0

3-SN DRAM size at each SN-F port	Top address bits[bit 1, bit 0]	Inv_top_address_bit value
2GB, total 6GB	[35, 31]	0b0
4GB, total 12GB	[33, 32]	0b0
8GB, total 24GB	[34, 33]	0b0
16GB, total 48GB	[39, 34]	0b0
32GB, total 96GB	[39, 36]	0b1
64GB, total 192GB	[37, 36]	0b0
128GB, total 384GB	[38, 37]	0b0

The following shows the 6-SN DRAM size settings.

Table 3-17: 6-SN DRAM size settings

6-SN DRAM size at each SN-F port	Top address bits[bit 2, bit 1, bit 0]	Inv_top_address_bit value
1GB, total 6GB	[35, 31, 28]	0b0
2GB, total 12GB	[33, 32, 28]	0b0
4GB, total 24GB	[34, 33, 28]	0b0
8GB, total 48GB	[39, 34, 33]	0b0
16GB, total 96GB	[39, 36, 28]	0b1
32GB, total 192GB	[37, 36, 28]	0b0
64GB, total 384GB	[38, 37, 28]	0b0

The top address bit combinations for 5-SN mode are not available with the PDD memory map for all DDR size combinations. For specific memory maps, you must follow the 6-SN rules to achieve valid address bit combinations and contiguous SN addresses.

3.4.3.2 SN contiguous address spaces

The physical address bits must be connected to an SN-F for various configurations.

If all HN-Fs send their cache misses to a single SN-F, that SN-F sees the full address space. However, it is common for a system to have two or more SN-Fs, with each HN-F sending its cache misses to a single SN-F. In this scenario, each SN-F receives only part of the address space. SN-F typically removes one or more address bits to retain a contiguous address map. The full physical address is presented to each SN-F for every request so that any SN-F based memory protection logic can function. However, the actual mapping to RAM locations can be done with the modified address. The address modification depends on multiple factors:

- Number of HN-Fs in the cache group
- Number of SN-Fs in the cache group
- Which HN-Fs share SN-Fs

2ⁿ-SN address striping

The following table provides HN-F and SN-F combinations that are supported within a cache group, along with the address bits that should be removed.

Table 3-18: HN-F and SN-F combinations supported within a cache group

Number of HN-Fs	Number of SN-Fs	Bits to strip from full PA
2	1	None
	2	[6]
4	1	None
	2	[7]
	4	[7, 6]
8	1	None
	2	[8]
	4	[8, 7]
	8	[8, 7, 6]
16	1	None
	2	[9]
	4	[9, 8]
	8	[9, 8, 7]
	16	[9, 8, 7, 6]
32	1	None
	2	[10]
	4	[10, 9]
	8	[10, 9, 8]
	16	[10, 9, 8, 7]
	32	[10, 9, 8, 7, 6]
64	1	None
	2	[11]
	4	[11, 10]
	8	[11, 10, 9]
	16	[11, 10, 9, 8]
	32	[11, 10, 9, 8, 7]

The method that is used to calculate the bits stripped is as follows:

1. The highest bit removed is the least significant address bit used in the XOR function for the most significant bit of the HN-F select hash function.

256 HN-Fs	PA[13]
128 HN-Fs	PA[12]
64 HN-Fs	PA[11]
32 HN-Fs	PA[10]
16 HN-Fs	PA[9]
Eight HN-Fs	PA[8]

Four HN-Fs	PA[7]
Two HN-Fs	PA[6]

- The number of bits stripped is $\log_2(\text{number of SN-Fs})$, sequentially below the highest bit.

This approach to bit stripping assumes that HN-Fs that share SN-Fs are sequential in the RN SAM cache group HN-F table. For example, if there are eight HN-Fs and two SN-Fs, the bottom four HN-Fs in the RN SAM table would share an SN-F. The top four HN-Fs would share an SN-F.

3-SN, 5-SN, and 6-SN address striping

As 3-SN, 5-SN, and 6-SN address hashing implements modulo function according to the top address bits used, the SN-F must remove these bits to achieve a contiguous memory map in the DRAM.

3-SN mode	The SN-F must remove top_address_bit1 and top_address_bit0.
5-SN mode	The SN-F must remove top_address_bit2, top_address_bit1, and top_address_bit0.
6-SN mode	The SN-F must remove top_address_bit2, top_address_bit1, and top_address_bit0.

3.4.3.3 Address bit masking in the HN-F SAM

CMN-700 supports masking of address bits in the HN-F SAM. Certain restrictions apply to this process.

Range-based comparison

HN-F SAM uses address bits:

- [MSB:26] when the applicable `*RCOMP_EN` parameter is 0.
- [MSB:2^{*RCOMP_LSB}] When the applicable `*RCOMP_EN` parameter is 1

By programming select bits to 0b0 in the `hnf_sam_region_cmp_addr_mask_reg` mask register, the incoming address and the programmed address ranges can be masked off before comparison. This region mask is only applicable to region-based memory partitioning in the HN-F and so the mask is not applied to the hashing scheme in 3-SN, 5-SN, and 6-SN modes.

Stripe function

HN-F SAM supports masking of address bits used for 3-SN, 5-SN, or 6-SN address hashing. This feature can be enabled by programming the `hn_sam_hash_addr_mask_reg`.

The following limitations apply:

- Range compare mask must not mask off bits that represent the size of the region. For example, if any of the region sizes are 64MB, address bit 26 cannot be masked. Similarly, if a region size is 512MB, address bit 29 cannot be masked.
- If 3-SN, 5-SN, and 6-SN mode is enabled, a configurable group of nine address bits in the range of address bits [21:8] and the configured `top_addr_bits` are essential in distributing the

addresses between memory. Arm recommends that these bits are masked carefully to avoid memory aliasing. See [3.4.3.1 HN-F to SN-F memory striping in HN-F SAM](#) on page 59

- The address bits masked in HN-F and RN SAM must be consistent. This requirement ensures that PrefetchTarget requests from an RN-F to SN-F are addresses by the same SN-F as SLC miss from the HN-F to SN-F.

3.4.3.4 HN-F SLC and SF flexible addressing

CMN-700 HN-F supports flexible Set and Tag address for both SLC and SF. This allows for custom hashing or sliding of *Physical Address* (PA) bits to index into the SLC and SF cache. It is also useful when RNSAM's hierarchical hashing scheme is used because it can lead to unused sets in SLC and SF in certain use cases.

HN-F provides one 64-bit flex register each for SLC and SF which can be used for programming various modes and masks in custom implementations.

The default implementation of HN-F uses consecutive setaddr bits starting from bit 6 of the PA. For example, a system with 52 bits of PA and the SLC containing 1024 sets requires 10 bits to address into the cache. So the setaddr bits are PA[15:6]. The remaining address bits PA[51:16] are saved in the SLC RAM as "tag".

Generally, the contents of PA for the default set/tag addressing is:

$$\{\text{TAG}, \text{SETADDR}, \text{LINEOFFSET}\} \leq \text{PA}[\text{PA_WIDTH_PARAM}-1:0]$$

Similarly, for SLC/SF victims, the PA is generated as an inverse of the previous equation (cache line aligned):

$$\text{PA}[\text{PA_WIDTH_PARAM}-1:0] \leq \{\text{TAG}, \text{SETADDR}, 6'b00\}$$

For information about modifying the Setaddr and Tag contents, see *Arm® Neoverse™ CMN-700 Coherent Mesh Network Configuration and Integration Manual*.

The HN-F flexible Set and Tag address supports two modes of programming:

- Sliding Set Address mode:

In this mode, set address start bit is configurable to accommodate hierarchical hashing in RNSAM (cluster mask is set to 64bytes). Set address start bit is programmed based on the number of clusters enabled in the Hierarchal hashing based on below equation:
Set address start bit = 6 + log2 (number of clusters)

When this mode is enabled, the SLC/SF setaddr is shifted as shown below. All remaining bits are used as Tag bits stored in the SLC/SF cache.

$$\text{PA}[\text{PA_WIDTH_PARAM}-1:0] = \{\text{TAG1}, \text{SETADDR}, \text{TAG0}, 6'b00\}$$

TAG = {TAG1, TAG0}, TAG0 bits are based on number of clusters

- Programming registers: (For a 4 cluster example, Set address start bit = 8)

- `cmn_hns_pa2setaddr_sf.setaddr_startbit_sf = 4'b1000;`
 - `cmn_hns_pa2setaddr_slc.setaddr_startbit_slc = 4'b1000;`
 - `por_mxp_p#{index}_pa2setaddr_slc.setaddr_startbit_slc_p#{index} = 4'b1000;`
 - `por_mxp_p#{index}_pa2setaddr_sf.setaddr_startbit_sf_p#{index} = 4'b1000;`
- Shuttering Set Address Mode

Alternately HN-F flexible Set address bits are selected by shuttering precise bit locations in the PA. This controllable selection is required to enable RNSAM hierarchical cluster mask capability. The shuttering scheme must only have one group of repetitive bits shuttered.

`PA[PA_WIDTH_PARAM-1:0] = {TAG1, SETADDR1, TAG0 (shuttered bits), SETADDR0, 6'b00}`

`SETADDR = {SETADDR1, SETADDR0}`, SETADDR0 bits are based on cluster mask.

`TAG = {TAG1, TAG0}`, TAG0 bits are based on number of clusters

- Programming registers: (For a given system, Cluster interleaving = 4KB & Clusters# = 8)
 - `cmn_hns_pa2setaddr_slc.setaddr_shutter_mode_en = 1'b1`
 - `cmn_hns_pa2setaddr_slc.setaddr_indx_[0-5]_shutter = 3'b000 //pass-through`
 - `cmn_hns_pa2setaddr_slc.setaddr_indx_[6-12]_shutter = 3'b011 // shift by 3 bits`
 - `cmn_hns_pa2setaddr_sf.setaddr_shutter_mode_en = 1'b1`
 - `cmn_hns_pa2setaddr_sf.setaddr_indx_[0-5]_shutter = 3'b000 //pass-through`
 - `cmn_hns_pa2setaddr_sf.setaddr_indx_[6-12]_shutter = 3'b011 // shift by 3 bits`
 - repeat above programming in `por_mxp#{index}_pa2setaddr_slc` and `por_mxp#{index}_pa2setaddr_sf` registers as well

1. Index ID in the programmable registers represents `setaddr[12:0]`
2. If the Shutter bit is higher than the Set address index, then the shuttering mode must not be enabled
 $\text{Set_Addr_width} = \log_2 ((\text{CACHE_SIZE in MB}) / (\text{Num_ways} * 64))$



Note

So the shuttering must only be enabled if `shutter_bit <= (set_addr_width + 6)`

3. In the `setaddr_indx` programming, once an index is programmed, all subsequent index must also have the same programming value. In the example 4KB interleaving, `setaddr_indx_[12:6]_shutter` must all be programmed to `3'b011`.
4. These registers are present in both HN-F and MXP_port connected to the HN-F. The programming for `setaddr` must be the same in MXP and HN-F

Table 3-19: HN-F SAM parameters and configuration registers

HN-F SAM Feature	Applicable Parameters	Applicable configuration registers
Global	HNSAM_RCOMP_EN	hn_sam_region_cmp_addr_mask_reg
	HNSAM_RCOMP_LSB	
Default hashed	HNSAM_DEF_HASHED_GRP_EN	cmn_hns_sam_control cmn_hns_sam_6sn_nodeid cmn_hns_sam_sn_properties cmn_hns_sam_sn_properties1 cmn_hns_sam_sn_properties2
Non-hashed regions	HNSAM_NUM_NONHASH	cmn_hns_sam_memregion0-1 cmn_hns_sam_memregion0-1_end_addr cmn_hns_sam_nonhash_cfg1_memregion2-63 cmn_hns_sam_nonhash_cfg2_memregion2-63 cmn_hns_sam_sn_properties cmn_hns_sam_sn_properties1
Hashed regions	HNSAM_NUM_HTG	hn_sam_hash_addr_mask_reg cmn_hns_sam_htg_cfg1_memregion0-15 cmn_hns_sam_htg_cfg2_memregion0-15 cmn_hns_sam_htg_cfg3_memregion0-15 cmn_hns_sam_htg_sn_nodeid_reg0-15 cmn_hns_sam_htg_sn_attr0-15
Aggregated SA support	-	cmn_hns_sam_ccg_sa_nodeid_reg0-3 cmn_hns_sam_ccg_sa_attr0-3
CPAG support	-	cmn_hns_cml_port_aggr_grp0-31_add_mask cmn_hns_cml_port_aggr_grp_reg0-12 cmn_hns_cml_port_aggr_ctrl_reg cmn_hns_cml_port_aggr_ctrl_reg1-6
Custom hash	HNSAM_CUSTOM_REGS	hnf_generic_regs0-7



Note

- Parameters and registers in the same row are not necessarily related
- Ranges in register names, m-n, indicates the register repeats (m - n + 1) times

3.4.4 SAM programming examples

This section contains the programming examples of the different devices supported by the CMN-700

3.4.4.1 RN SAM programming with power-of-two hashing

This is an example for a system with 32-HNFs, without CAL, HN SAM uses 3-SN or Direct-mapping mode.

Hashing function:

```
Select[4:0] = { (10^15^20^...^50),
(9^14^19^...^49),
(8^13^18^...^48),
(7^12^17^...^47),
(6^11^16^...^51) }
```

Example 3-2: RN SAM programming

```
#####
# Set up SCG0
#####
sys_cache_grp_region0.region0_valid = 1'b1;          # Set SCG0 Valid
sys_cache_grp_region0.region0_target_type = 3'b000;   # target type HNF
sys_cache_grp_region0.region0_base_addr = <base address for SCG>;
                                                    # SCG0 Base Address
sys_cache_grp_region0.region0_size = <size of the SCG>; # SCG0 size
sys_cache_group_hn_count.scg0_num_hnf = 8'h20;        # 32 HNs in SCG0

#####
# Program the 32 HNF targetIDs -
# sys_cache_grp_hn_nodeid_reg[7:0] covers 32 HNF targetIDs for cluster0
#####
for (indx=0, indx=indx+1, indx <8) begin          # 8 regs for 32 HNF nodeIDs
    for (id=0, id=id+1, id <4) begin                # each reg covers 4 nodeIDs
        sys_cache_grp_hn_nodeid_reg#{indx}.nodeid_{4*indx + id} = <HNF#{4*indx +
id}targetID>
    end
end

#HNSAM programming
#####
# 3-SN mode
#####

#Program each HNF to 3-SN mode
cmn_hns_sam_control.hn_cfg_three_sn_en = 1'b1;
cmn_hns_sam_6sn_nodeid.hn_hash_addr_bits_sel = 3'b001;
cmn_hns_sam_control.hn_cfg_sam_top_address_bit0 = 6'd32;
                                                    # depends on customer memory map
cmn_hns_sam_control.hn_cfg_sam_top_address_bit1 = 6'd33;
                                                    # depends on customer memory map
```

```
cmn_hns_sam_control.hn_cfg_sn0_nodeid = < sn0_nodeid>
cmn_hns_sam_control.hn_cfg_sn1_nodeid = < sn1_nodeid>
cmn_hns_sam_control.hn_cfg_sn2_nodeid = < sn2_nodeid>

#####
# OR
#####

#####
# Direct Mapping mode
#####
# Program each HNF to Direct Mapping mode
# With 32 HNFs target 4 SNs

for (hnf_num=0, hnf_num = hnf_num +1, hnf_num <32) begin          # begin
    # Group 8 HNFs and assign 1 SN target for all HNFs in that group
    cmn_hns_sam_control.hn_cfg_sn0_nodeid = <sn#{hnf_num%8}_nodeid>
end
```

3.4.4.2 RN SAM programming for Prefetch Target to a non-hash region

The following is an example for programming the RN SAM for a Prefetch Target to a non-hash region.

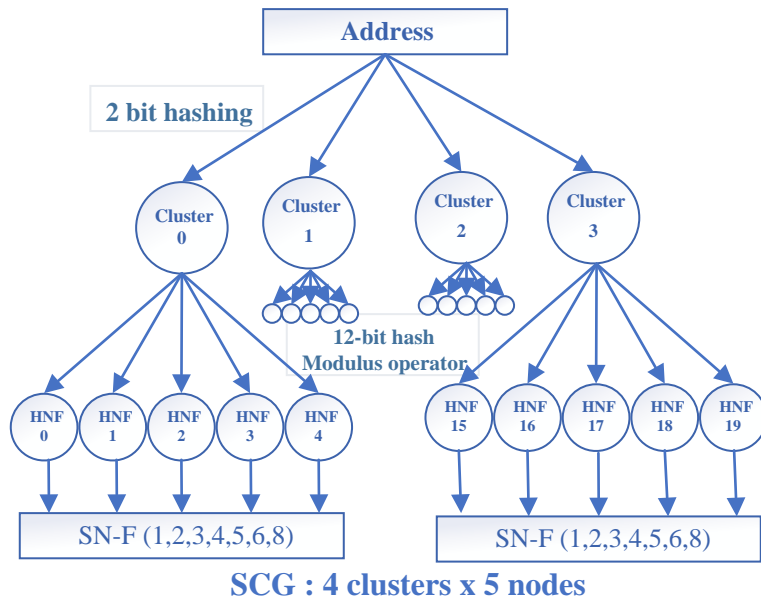
Example 3-3: RN SAM programming for a Prefetch Target

```
# Non-hashed region programming for SCG0 (2-129)
sam_scg0_prefetch_nonhashed_mem_region_cfg1_reg0.
    scg0_prefetch_nonhash_reg0_region_valid = 1'b1;
sam_scg0_prefetch_nonhashed_mem_region_cfg1_reg0.
    scg0_prefetch_nonhash_reg0_base_address = <start address of the region>;
sam_scg0_prefetch_nonhashed_mem_region_cfg2_reg0.
    scg0_prefetch_nonhash_reg0_end_address = <end address of the region>;
sam_scg0_prefetch_nonhashed_mem_region_cfg2_reg0. scg0_prefetch_nonhash_reg0_tgtid =
    <SN nodeID>
```

3.4.4.3 CMN-700 SAM programming for Hierarchical Hashing

This is an example of Hierarchical hashing of a CMN-700 system configured for 4 clusters, 32 HNFs with CALs, and HNSAM in 3-SN mode.

Figure 3-18: Hierarchical hashing configuration



- First hierarchy of hashing (clusters) is only power of two (supported clusters #2,4,8,16,32).
- Second hierarchy of hashing (nodes) is non-power of two (supported nodes # [2-32]).

First-level Hierarchy hashing:

Hashing for four clusters:

- Number of Address bits in select: 2
- $\text{select}[0] = (6^8 10^{\dots} 50)$
- $\text{select}[1] = (7^9 11^{\dots} 51)$

Second-level Hierarchy hashing

12 bit hashing:

- $\text{hash12}[11:0] = \{2'b00, \text{Addr}'[51:42]\} \wedge \text{Addr}'[41:30] \wedge \text{Addr}'[29:18] \wedge \text{Addr}'[17:6]$
- $\text{select}[6:0] = (\{ \text{hash12}[11] \wedge \text{hash12}[0],$

$\text{hash12}[10] \wedge \text{hash12}[1],$

$\text{hash12}[9] \wedge \text{hash12}[2],$

$\text{hash12}[8] \wedge \text{hash12}[3],$

$\text{hash12}[7] \wedge \text{hash12}[4],$


```
hash12[6] ^ hash12[5],

hash12[5:0] } * num_hnf ) >> 12
```

Example 3-4: RNSAM programming with CAL

For each RNSAM:

```
#Hierarchical hashing config:
sys_cache_grp_region0.region0_base_addr = <base address for SCG>
                                                    # Set HTG0 Base Address
sys_cache_grp_region0.region0_valid = 1'b1;
                                                    # Set HTG0 Valid
hashed_tgt_grp_cfg2_region0.region0_end_addr = <end address for SCG>
                                                    # Set HTG0 End Address
hashed_target_grp_hash_cntl_reg0.htg_region0_hierarchical_hash_en = 1'b1;
                                                    # Set Hier hashing for HTG0
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_hash_clusters = 6'h4;
                                                    # 4 clusters in HTG0
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_hash_nodes = 6'h10;
                                                    # 16 nodes in a cluster
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_enable_address_stripping = 3'b010;
                                                    #strip bit[7:6] for 4
clusters
sys_cache_group_hn_count.scg0_num_hnf = 8'h40 ;
                                                    # 32 HNs in HTG0
sys_cache_grp_cal_mode_reg0.scg0_hnf_cal_mode_en = 1'b1
                                                    # enable CAL mode

#####
# With CAL2 program the 32 HNF targetIDs - 2 clusters - 16 targetIDs per cluster
# sys_cache_grp_hn_nodeid_reg[3:0] covers HNF targetIDs 15:0 for cluster0
# sys_cache_grp_hn_nodeid_reg[7:4] covers HNF targetIDs 31:16 for cluster1
# sys_cache_grp_hn_nodeid_reg[11:8] covers HNF targetIDs 47:32 for cluster2
# sys_cache_grp_hn_nodeid_reg[15:12] covers HNF targetIDs 63:48 for cluster3
#####
for (clustr=0, clustr = clustr+1, clustr<4) begin # 4 clusters with 16 HNF nodeIDs
each
for (indx=0, indx=indx+1, indx <4) begin # 4 regs for each cluster for 16 HNF
nodeIDs
for (id=0, id=id+1, id <4) begin # each reg covers 4 nodeIDs
# 64 HNF targetIDs (cluster0,1,2,3)
sys_cache_grp_hn_nodeid_reg#{4*clustr+indx}.nodeid #{16*clustr+4*indx+id}
= <HNF#{16*clustr+4*indx+id} {targetID[10:1],1'b0} >
# upper 64 HNF targetIDs(not required in CAL mode)
end
end
end
end

#####
# Prefetch SN targets in CAL mode
#####
sys_cache_grp_sn_attr.sn_hash_addr_bits_sel_sys_cache_grp0 = 3'b000;
sys_cache_grp_sn_attr.sn_mode_sys_cache_grp0 = 3'b001;
sys_cache_grp_sn_sam_cfg0.scg0_top_address_bit0 = 6'd33;
# depends on customer memory map
sys_cache_grp_sn_sam_cfg0.scg0_top_address_bit1 = 6'd34;
# depends on customer memory map
-----

# Total of 32 sys_cache_grp_sn_nodeid_regs available to program the SN targetIDs
# 1 sys_cache_grp_sn_nodeid_regs covers 4 SN targetIDs
# 4 sys_cache_grp_sn_nodeid_regs covers up to 16 SN targetIDs in a cluster
# (but don't require all regs)
```

```
# In 3-SN mode, just require reg0 to program 3-SN targetIDs (reg1-3 not required) in
cluster0.
# reg4, 8 and 12 cover HNs for cluster 1,2,3 respectively
# Cluster0 uses 4 HN regs to cover 16 HN TargetIDs, similarly 4
sys_cache_grp_sn_nodeid_regs
# are allocated for cluster0 .Since we have only 3 SNs in each cluster we program
only
# sys_cache_grp_sn_nodeid_reg0
-----
for (indx=0, indx=indx+4, indx<16) begin          # SN mapping for the 4 clusters
    sys_cache_grp_sn_nodeid_reg#{indx}.sn_nodeid_0 = <clstr#{indx/4}_sn0>
    sys_cache_grp_sn_nodeid_reg#{indx}.sn_nodeid_1 = <clstr#{indx/4}_sn1>
    sys_cache_grp_sn_nodeid_reg#{indx}.sn_nodeid_2 = <clstr#{indx/4}_sn2>
end
```

Program each HN-F in a cluster to 3-SN mode:

```
cmn_hns_sam_control.hn_cfg_three_sn_en = 1'b1;
cmn_hns_sam_6sn_nodeid.hn_hash_addr_bits_sel = 3'b000;
cmn_hns_sam_control.hn_cfg_sam_top_address_bit0 = 6'd33;
                        # depends on customer memory map
cmn_hns_sam_control.hn_cfg_sam_top_address_bit1 = 6'd34;
                        # depends on customer memory map

cmn_hns_sam_control.hn_cfg_sn0_nodeid = <clstr_sn0_nodeid>
cmn_hns_sam_control.hn_cfg_sn1_nodeid = <clstr_sn1_nodeid>
cmn_hns_sam_control.hn_cfg_sn2_nodeid = <clstr_sn2_nodeid>
```

The following programming is required because of number of cluster(4) and hierarchical hashing:

```
# shift SLC/SF set address bits
cmn_hns_pa2setaddr_sf.setaddr_startbit_sf = 4'b1000;
                        # 4 clusters: Setaddr starts from PA[8]
cmn_hns_pa2setaddr_slc.setaddr_startbit_sf = 4'b1000;
                        # 4 clusters: Setaddr starts from
PA[8]
```

For each MXP connected to the HNF:

```
# shift SLC/SF set address bits
por_mxp_p#{index}_pa2setaddr_slc.setaddr_startbit_slc_p#{index} = 4'b1000;
                        # 4 clusters: Setaddr starts from PA[8]
por_mxp_p#{index}_pa2setaddr_sf.setaddr_startbit_sf_p#{index} = 4'b1000;
                        # 4 clusters: Setaddr starts from PA[8]
```

3.4.4.4 Programming sequence to enable CXL.mem regions inside HNSAM

```
# To define CXL memory regions, we must enable the following HNSAM user parameters:
# POR HNSAM NUM HTG PARAM = 2 (two regions)
# Region0 (HTG0) for 2 CCGs, CCG0, CCG1
# For each HNF:
cmn_hns_sam_htg_cfg1_memregion0.htg_region_valid0 = 1'b1;
cmn_hns_sam_htg_cfg1_memregion0.htg_region_base_addr0 = <start address of the
region>;
cmn_hns_sam_htg_cfg2_memregion0.htg_region_end_addr0 = <end address of the region>;
cmn_hns_sam_htg_cfg3_memregion0.htg0_sn_mode = 3'b111;
                        (Aggregated CCG SA selection
function)
```

```
cmn_hns_sam_htg_cfg3_memregion0.htg0_sa_device_interleave_cntl = 4'b0000;
                                                                (64 byte
interleaved)
cmn_hns_sam_htg_cfg3_memregion0.htg0_sa_ports_cnt = 3'b001      (2 CXL links)
cmn_hns_sam_ccg_sa_nodeid_reg0.cxg_sa_nodeid_0 = <CCG nodeID>
cmn_hns_sam_ccg_sa_nodeid_reg0.cxg_sa_nodeid_1 = <CCG nodeID1>
```

For one CXL link, we can either use HNSAM HTG or HNSAM non-hashed region:

```
# Region1 Hashed HTG1 for CCG2 - HTG programming
# For each HNF:
cmn_hns_sam_htg_cfg1_memregion1.htg_region_valid1 = 1'b1;
cmn_hns_sam_htg_cfg1_memregion1.htg_region_base_addr1 = <start address of the
region>;
cmn_hns_sam_htg_cfg2_memregion1.htg_region_end_addr1 = <end address of the region>;
cmn_hns_sam_htg_cfg3_memregion1.htg1_sn_mode = 3'b111;
                                                                (Aggregated CCG SA selection
function)
cmn_hns_sam_htg_cfg3_memregion1.htg1_sa_device_interleave_cntl = 4'b0000;
                                                                (64 byte
interleaved)
cmn_hns_sam_htg_cfg3_memregion1.htg1_sa_ports_cnt = 3'b000      (1 CXL links)
cmn_hns_sam_ccg_sa_nodeid_reg0.cxg_sa_nodeid_2 = <CCG nodeID>

# OR

# Region1 - non-hashed for CCG2 - Non-hashed region programming
# Foreach HNF:
cmn_hns_sam_memregion0.valid = 1'b1;
cmn_hns_sam_memregion0.base_addr = <start address of the region>;
cmn_hns_sam_memregion0_end_addr.end_addr = <end address of the region>;
cmn_hns_sam_memregion0.range0_nodeid = <CCG nodeID>
```

3.4.4.5 Multichip flat, 64B interleaving across 4 chips: (tgt_type = HN-Fs)

32 HNF's per chip, 128HNF across 4 chips, 2 CCGs connecting any two chips

Example 3-6: RNSAM programming (without CAL):

For each RNSAM:

```
sys_cache_grp_region0. region0_base_addr = <base address for SCG>
                                                                # Set HTG0 Base Address
sys_cache_grp_region0. region0_valid = 1'b1;
                                                                # Set HTG0 Valid
hashed_tgt_grp_cfg2_region0. region0_end_addr = <end address for SCG>
                                                                # Set HTG0 End Address
sys_cache_group_hn_count. scg0_num_hnf = 8'h80 ;
                                                                # 128 HNs in HTG0
sys_cache_grp_cal_mode_reg0.scg0_hnf_cal_mode_en = 1'b0
                                                                # do not enable CAL mode

#####
# Program the 128 HNF targetIDs
• HNF node id's on the local chip are programmed.
• HNF node id's on the remote chips are don't care (as these are overridden by CCG
nodes).
• Set cpa_en = 1'b1 for the remote HNF's and program the cpag id for the remote
HNF's.
#####
for (indx=0, indx=indx+1, indx <16) begin
                                                                # 32 registers for all the
HNF's
```

```

for (id=0, id=id+1, id <4) begin                                # 4 HNF's per each register
    sys_cache_grp_hn_nodeid_reg#{indx}. nodeid #{4*indx + id} = <HNF#{4*indx + id}>
    hashed_target_grp_hnf_nodeid_reg#{16 + indx}.nodeid_# {64 + 4*indx + id}
    = <HNF#{64 + 4*indx + id}>
end
end

sys_cache_grp_hn_cpa_en_reg.hash_cpa_en = 64'hEEEEEEEEEEEEEEEE
                                     (cpa enable for 64 HNF's (0 - 63))
hashed_target_grp_hnf_cpa_en_reg1. htg_hnf_cpa_en1 = 64'hEEEEEEEEEEEEEEEE
                                     (cpa enable for 64 HNF's (64 -
127))
sys_cache_grp_hn_cpa_grp_reg.enable_multi_cpa_grp_scg0 = 1'b1

for (indx=0, indx=indx+1, indx <16) begin                      # 16 registers (16 x 8 = 128)
    for (id=0, id=id+1, id <8) begin                            # 8 CPAG ids for each register
        hashed_target_grp_cpag_perhnf_reg{indx}.htg_cpag_hnf{indx*8 + id} = <CPAG for
        HNF{indx*8 + id}>
    end
end

## program the CPAG's

cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_grp1_add_mask = <addr mask for the CPAG1>
cml_port_aggr_grp2_add_mask = <addr mask for the CPAG2>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.cpag_valid1 = 1'b1
cml_port_aggr_ctrl_reg.cpag_valid2 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_ctrl_reg.num_cxg_pag1 = 3'b001 (2 CCG ports)
cml_port_aggr_ctrl_reg.num_cxg_pag2 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0. pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0. pag_tgtid1 = <CCG-1>
cml_port_aggr_grp_reg0. pag_tgtid2 = <CCG-2>
cml_port_aggr_grp_reg0. pag_tgtid3 = <CCG-3>
cml_port_aggr_grp_reg0. pag_tgtid4 = <CCG-4>
cml_port_aggr_grp_reg1. pag_tgtid4 = <CCG-5>

```

3.4.4.6 PCIe IO Traffic examples (RN-I to HN-F)

The following table describes the PCIe IO traffic examples for local and remote interleaving.

Table 3-20: PCIe IO traffic interleaving examples

Traffic Examples	Local vs remote	Interleave granularity	RN SAM config
3.4.4.6.1 HNF interleave across local/remote HNF at cache line interleaving of 64bytes on page 76	local/remote flat	<4K	HTG (addr based) CPAG (axid)
3.4.4.6.2 HNF interleave across local/remote HNF: 4K interleaving across chips and 64 bytes interleaving across HNFs on page 77	local/remote flat	>=4K	HTG (addr based) CPAG (axid)
3.4.4.6.3 Remote NUMA (non-hash region to CPAG) on page 78	remote NUMA	NA	non-hash CPAG (axid/addr)

3.4.4.6.1 HNF Interleave across local/remote HNF at cache line interleaving of 64bytes

The following example is of two chips with 16 HN-Fs on each, using simple power of two hashing.

Example 3-7: Enabling Interleave

```
#####
# Set up SCG0
#####
sys_cache_grp_region0. region0_valid      = 1'b1;
                                # Set SCG0 Valid
sys_cache_grp_region0. region0_target_type = 3'b000;
                                # target type HNF
sys_cache_grp_region0. region0_base_addr  = <base address for SCG>;
                                # SCG0 Base Address
sys_cache_grp_region0. region0_size       = <size of the SCG>;
                                # SCG0 size
sys_cache_group_hn_count. scg0_num_hnf    = 8'h20;
                                # 32 HN's in SCG0

#####
# Program the 32 HNF targetIDs -
# sys_cache_grp_hn_nodeid_reg[7:0] covers 32 HNF targetIDs for cluster0
#####
for (indx=0, indx=indx+1, indx <8) begin      # 8 regs for 32 HNF nodeIDs
    for (id=0, id=id+1, id <4) begin          # each reg covers 4 nodeIDs
        sys_cache_grp_hn_nodeid_reg#{indx}. nodeid_{4*indx + id} = <HNF#{4*indx + id}
        targetID >
    end
end

sys_cache_grp_hn_cpa_en_reg.hash_cpa_en = 32'hCCCCCCCC
                                (cpa enable for 32 HNF's (0 - 31))
sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg0 = <CCG0>

## program the CPAG's
cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0. pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0. pag_tgtid1 = <CCG-1>
```

3.4.4.6.2 HNF Interleave across local/remote HNF: 4K interleaving across chips and 64 bytes interleaving across HN-Fs

The following example is of two chips with 16 HN-Fs on each. To enable 4K interleaving across local and remote chips, you must enable Hierarchical hashing and Cluster mask.

Example 3-8: Hierarchical hashing config

```
sys_cache_grp_region0. region0_base_addr = <base address for SCG>
                                # Set HTG0 Base Address
sys_cache_grp_region0. region0_valid = 1'b1;
                                # Set HTG0 Valid
hashed_tgt_grp_cfg2_region0. region0_end_addr = <end address for SCG>
                                # Set HTG0 End Address
hashed_target_grp_hash_cntl_reg0. htg_region0_hierarchical_hash_en = 1'b1;
                                # Set Hier hashing for HTG0
hashed_target_grp_hash_cntl_reg0. htg_region0_hier_hash_clusters = 6'h2;
                                # 2 clusters in HTG0 (2 chips)
hashed_target_grp_hash_cntl_reg0. htg_region0_hier_hash_nodes = 6'h10;
```

```

# 16 nodes in a cluster
hashed_target_grp_hash_cntl_reg0.htg_region0_hier_enable_address_stripping = 3'b001;

#strip bit[6] for 2 clusters
hashed_target_grp_hash_cntl_reg0.hier_cluster_mask = 4'b0110;
# 4K interleaving
sys_cache_group_hn_count.scg0_num_hnf = 8'h20 ;
# 32 HNs in HTG0

for (clustr=0, clustr = clustr+1, clustr<2) begin # 2 clusters with 16 HNF nodeIDs
    each
    for (indx=0, indx=indx+1, indx <4) begin # 4 regs for each cluster for 16
        HNF nodeIDs
        for (id=0, id=id+1, id <4) begin # each reg covers 4 nodeIDs
            # 32 HNF targetIDs (cluster0,1)
            sys_cache_grp_hn_nodeid_reg#{4*clustr+indx}.nodeid_# {16*clustr+4*indx+id }
            = <HNF#{16*clustr+4*indx+id} targetID >
        end
    end
end

sys_cache_grp_hn_cpa_en_reg.hash_cpa_en = 32'hFFFF0000
# (cpa enable for 32 HNF's (0 - 31))
sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg0 = <CCG0>

## program the CPAG's
cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0.pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0.pag_tgtid1 = <CCG-1>
cml_port_aggr_grp_reg0.cpag_axid_hash_en = 1'b1
    
```

3.4.4.6.3 Remote NUMA (non-hash region to CPAG)

The following example is of two chips with 16 HN-Fs on each, where the HN-Fs on the remote chip are configured as NUMA through the non-hashed region and CPAG (axid based hashing).

Example 3-9: Configuring non-hash region

```

non_hash_mem_region_reg0.base_addr = <base addr>
non_hash_mem_region_reg0.valid = 1'b1
non_hash_mem_region_cfg2_reg0.end_addr= <end addr>
cml_port_aggr_mode_ctrl_reg.cpag_en = 1'b1
cml_port_aggr_mode_ctrl_reg.cpag_grpid = CPAG0

# program the CPAG's
cml_port_aggr_grp0_add_mask = <addr mask for the CPAG0>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 3'b001 (2 CCG ports)
cml_port_aggr_grp_reg0.pag_tgtid0 = <CCG-0>
cml_port_aggr_grp_reg0.pag_tgtid1 = <CCG-1>
cml_port_aggr_grp_reg0.cpag_axid_hash_en = 1'b1
    
```

3.4.4.6.4 DSU HN-I programming example

The following example shows how to program the DSU for the HN-I in CMN-700

Example 3-10: DSU HN-I programming

```
dsu_hni_region_reg.dsu_hni_base_addr = <start addr>
# Base address must be power of two size aligned.
# Total size is decided by number of DSUs and size of each DSU HNI.
# Number of DSU's is a build time parameter.

dsu_hni_region_reg.dsu_hni_region_size = < size of the each HNI region>
dsu_hni_region_reg.dsu_hni_region_valid = 1'b1;
```

3.4.5 Cross chip routing and ID mapping

IDs are generated and used to route CML_SMP protocol messages across multiple chips.

The following acronyms are used in this section:

- *Request Agent ID* (RAID)
- *Home Agent ID* (HAID)
- *Logical Device ID* (LDID)

By default, LDIDs are uniquely assigned within a device type. For example, RN-Fs in the system can be assigned LDIDs 0-n. RN-Is can be assigned LDIDs 0-m. RN-Ds can be assigned LDIDs 0-k.

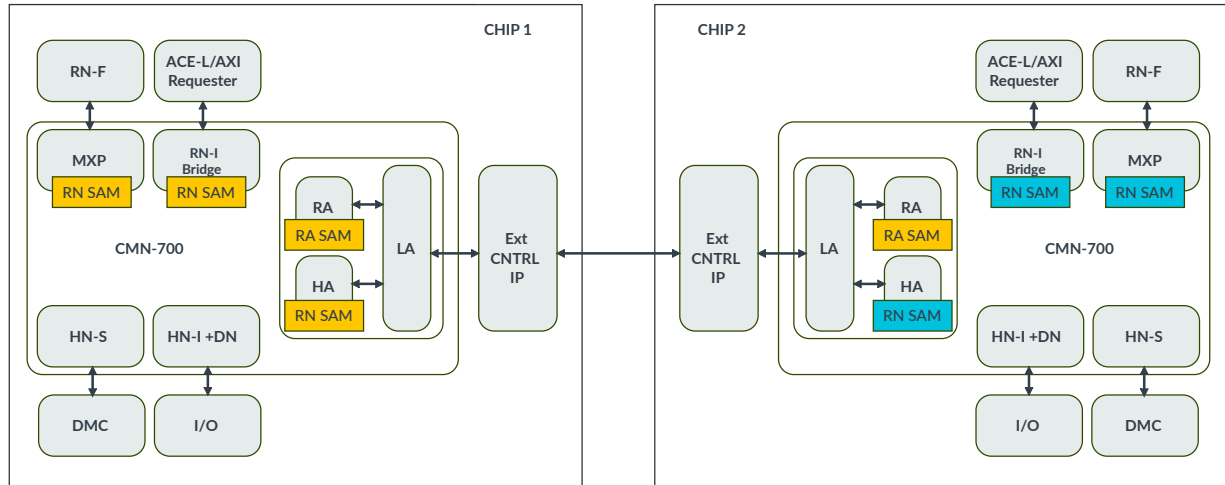
The following rules apply to CML configurations with HN-F SF clustered mode disabled:

- RAID usage:
 - Confined to CML gateway devices only.
 - All local and remote components that are visible to CMN-700 use and operate on sequentially assigned LDIDs. CCG devices bidirectionally map each RAID to an LDID.
- RN-F default LDID assignment:
 - Local RN-Fs are assigned LDIDs from 0-n, sequentially
 - Remote RN-Fs must be assigned LDIDs n+1 and above by the discovery software

If HN-F SF clustered mode is enabled and LDID override is used, the preceding rules do not apply. In this mode, local and remote RN-Fs can be assigned LDIDs from the full ID space to meet the clustering requirement. The number of bits required for RN-F LDID values, the LDID_WIDTH, is determined by the total number of local and remote RN-F/RA nodes and determines the full ID space.. To maximize SF efficiency, assigned LDIDs must be sequential and the LDID space must not have any holes.

The following figure shows a basic multi-chip block diagram.

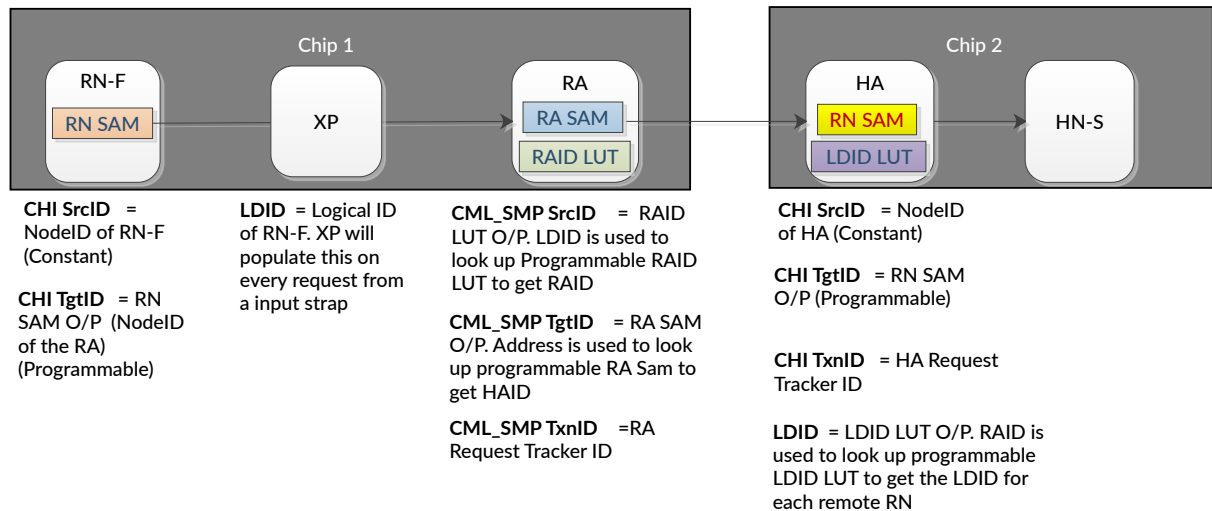
Figure 3-19: Multi-chip block diagram



Request from an RN-F to a remote HN-F

The following figure shows all IDs generated and used to route a request from a local RN-F on chip 1 to a remote HN-F on chip 2.

Figure 3-20: RN-F to remote HN-F IDs



The flow for this process is as follows:

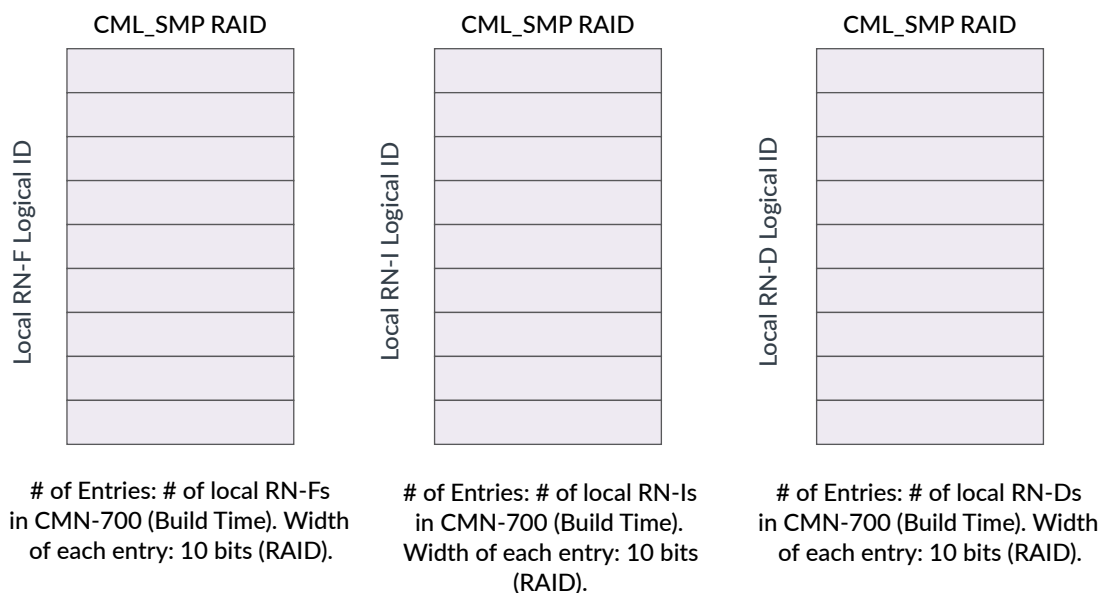
- The RN-F looks up the programmable RN SAM to populate the CHI target ID on a request.
- The XP populates the RN-F LDID on this request.
- When HN-F SF clustering is disabled, the LDIDs for local RN-Fs must not be changed. The build-time LDID assignments are discovered by reading any one of the HN-F

cmn_hns_rn_cluster<X>_physid<Y> registers. A few cycles after reset, these registers are prepopulated with the LDIDs for local RN-Fs within that chip.

- If HN-F SF clustered mode is enabled:
 - The local RN-F LDID in por_mxp_p[0-5]_ldid_override register can be programmed to match the clustering requirements at each RN-F port. In clustered mode, the local RN-F LDIDs are not pre-programmed in the physical ID registers of the HN-F out of reset. Therefore, you must explicitly program these registers to suit the clustering requirements.
 - When HN-F SF clustered mode is enabled, CMN-700 allows override of the LDID of each RN-F. This LDID must be programmed in the registers of the CML-RA. Also, each override value must match the por_mxp_p[0-5]_ldid_override register value of the corresponding RN-F.
 - If HN-F SF clustered mode is enabled, there must be at least two cluster groups. In other words, you cannot cluster all RN-Fs into a single cluster group.
 - If HN-F SF clustered mode is enabled, and there is a single RN-F in the cluster, then it must always use the lowest device ID in that cluster. Consider a four-way clustering, where the device ID fields are 0b00, 0b01, 0b10, and 0b11. In this configuration, you cannot use device ID values of 0b01, 0b10, or 0b11 unless 0b00 is also in use.
- por_mxp_device_port_connect_ldid_info_p[0-5] captures the default LDID values assigned to the RN-Fs that are connected to the respective device port.
- The cmn_hns_rn_cluster_*_physid_reg* registers also contain fields to program the source type for each RN-F in the system. For all local RN-Fs, the source type must be programmed to the appropriate CHI protocol issue that the RN-F supports. For all remote RN-Fs, the source type must be programmed to 0b1100 (CHI-E) as CML-HA is a proxy for all remote RN-Fs.
- RN-Is and RN-Ds are internal to CMN-700 and get their logical ID assigned during CMN-700 generation. The RN-I or RN-D sends this LDID on every request.
- CML-RA contains programmable lookup tables, RAID LUTs, for each class of local RN (RN-F, RN-I, RN-D, and CML-HA). CML_SMP discovery software discovers all local RN-Fs, RN-Ds, and RN-Is, and programs their corresponding RAIDs in these LUTs.
- The LDID of the incoming request is used to look up these RAID LUTs and determine the CML SMP RAID. CML-RA also has CML RA SAM. This CML RA SAM is used to generate the HAID. This HAID is used as the target ID to route the CML request message.
- The build-time LDID assignment can be discovered by reading any of the por_ccg_ra_rnf_ldid_to_nodeid_reg registers.

The following figure shows the programmable registers during CML discovery.

Figure 3-21: Programmable registers during CML Discovery



The CML-HA contains a programmable register to program the local LDID for each remote CML RAID that communicates with local CHI HNs on a given chip or socket. Each entry in the programmable register also contains an RN-F bit to identify whether the remote CML-RA is a caching agent (RN-F) or not. If RN-I and RN-F have the same RAID, then you must only fill the RN-F details in the entry.

HN-Fs on the local chip use this LDID to track a line in its SF. Therefore a unique LDID assignment is required for each remote requesting caching agent. These unique LDIDs must not overlap with LDIDs assigned to local RN-Fs. It is assumed that these IDs are assigned after CML discovery is complete. For example, all the CML-RAs are discovered and assigned an RAID.

The following figure shows the programmable register for RAID to LDID during CML discovery.

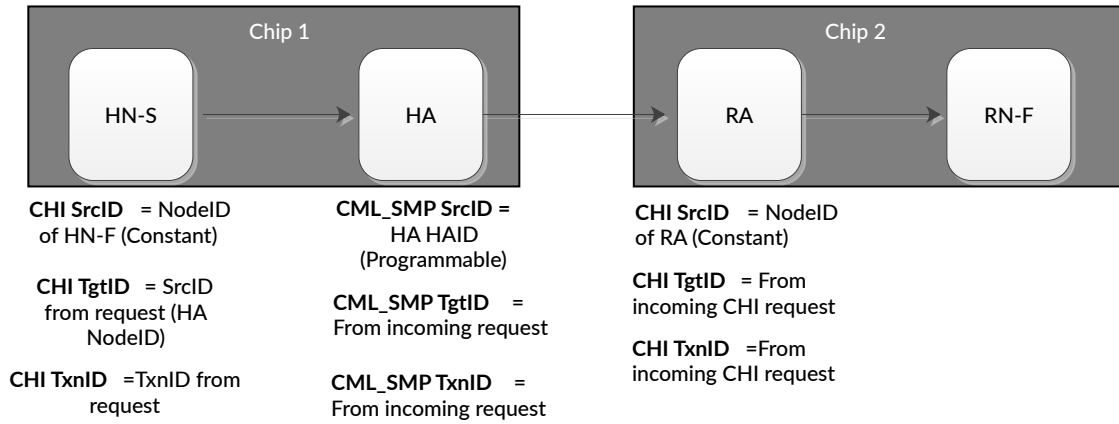
With the LDID passed to HN-F in all CHI REQ flits, HN-F uses this LDID as the true logical ID for SF tracking purposes. HN-F uses a logical ID vector in the SF. Depending on whether SF clustered mode is enabled or not:

- LDIDs can be uniquely tracked in the SF
- Multiple LDIDs can be aliased to a single logical ID

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

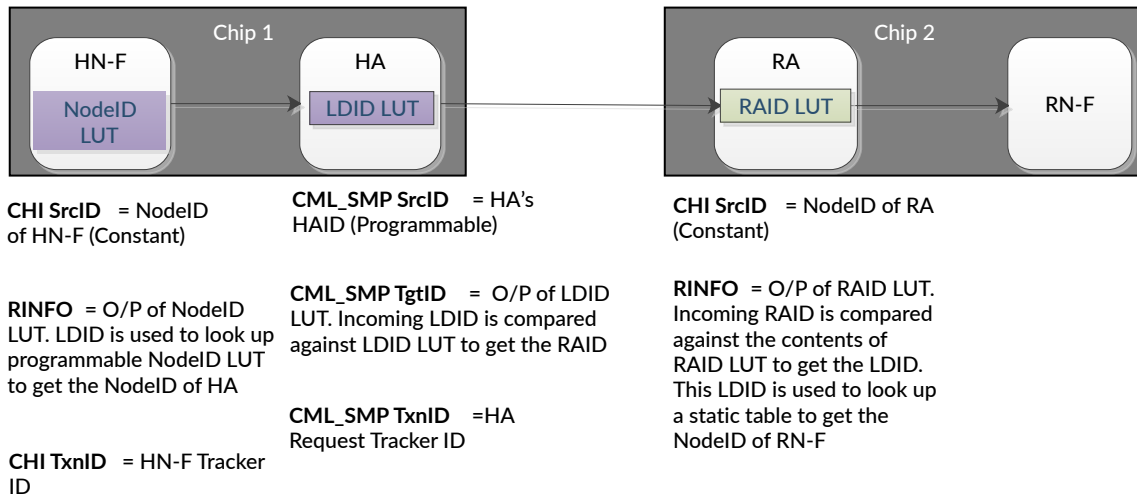
The following figure shows all generated IDs used to route a response from a remote HN-F on chip 2 to an RN-F on chip 1.

Figure 3-22: Remote HN-F to RN-F IDs



The following figure shows the flow of a snoop from an HN-F to a remote RN-F.

Figure 3-23: Snoop from HN-F to remote RN-F



The HN-F contains the following programmable LUT to program the CML-HA node ID of each remote caching agent. HN-F uses the unique LDID from the snoop vector to look up the physical CHI node ID of the CML-HA where snoops are sent to. The following table shows an example programming where logical IDs 0-7 are assigned to the local RN-Fs. The logical IDs 8-15 are assigned to the remote RN-Fs. Software assigns these IDs during the discovery process.

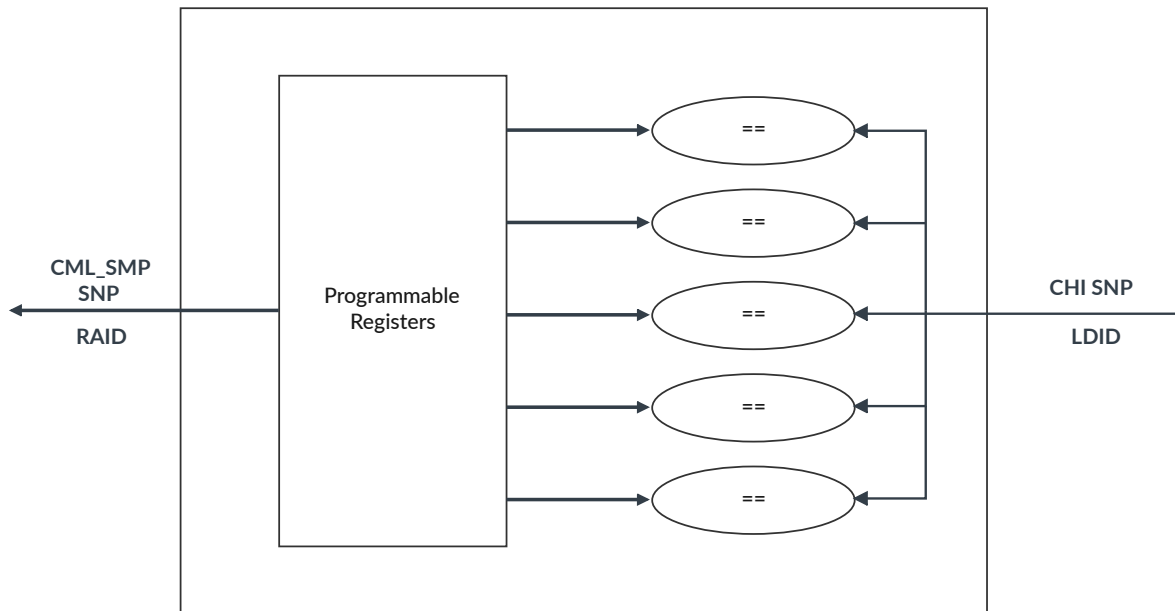
Table 3-21: Example program

Logical ID as index	HN-F programmable register	
	CHI node ID	ID valid
0	Local RN-F 0	1

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

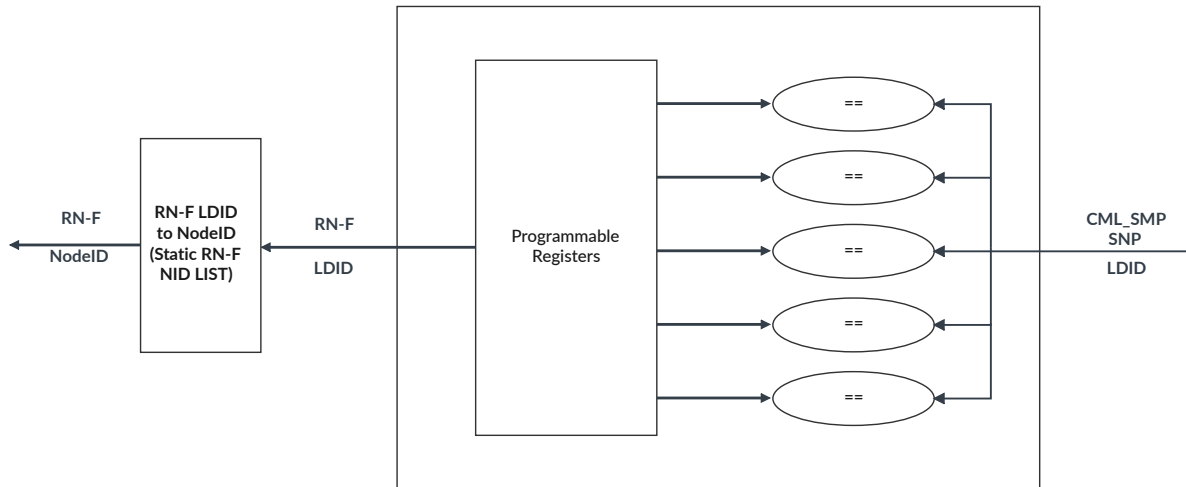
The CML-HA uses the LDID from incoming CHI snoop to perform a content match against the entries of programmable CML RAID to local LDID LUT. This content match results in the CML RAID sending a CML SMP snoop, as the following figure shows.

Figure 3-24: CHI SNP LDID to CML SNP RAID flow



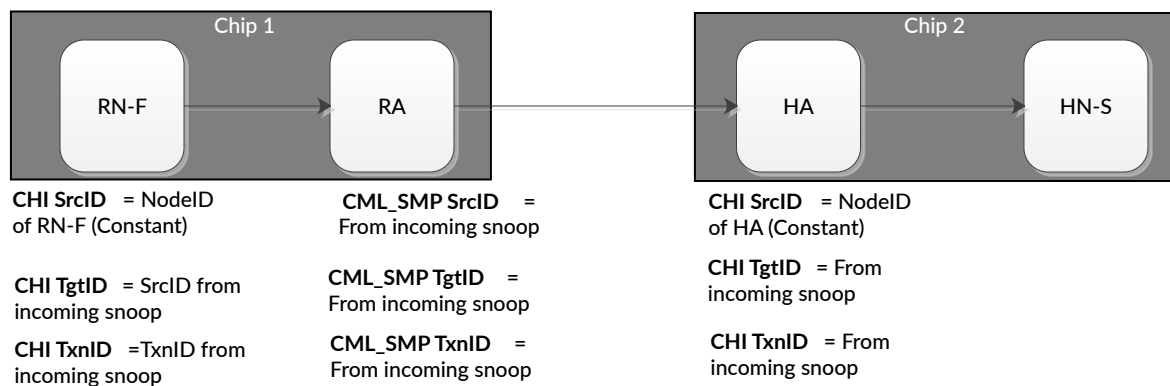
The following figure shows the detailed flow of a CHI SNP LDID to CML SNP RAID conversion.

Figure 3-25: CML SNP RAID to CHI RN-F LDID flow



The following figure shows all generated IDs used to route a snoop response from a remote RN-F on chip 1 to an HN-F on chip 2.

Figure 3-26: Remote RN-F to HN-F with all IDs generated



3.5 Reliability, Availability, and Serviceability

The CMN-700 *Reliability, Availability, and Serviceability* (RAS) features are implemented as set of distributed logging and reporting registers and a central interrupt handling unit.

The distributed logging and reporting registers are associated with devices that can detect errors. These devices are XP, HN-I, HN-F, SBSX, and CCG. The central interrupt handling unit is located in the HN-D.

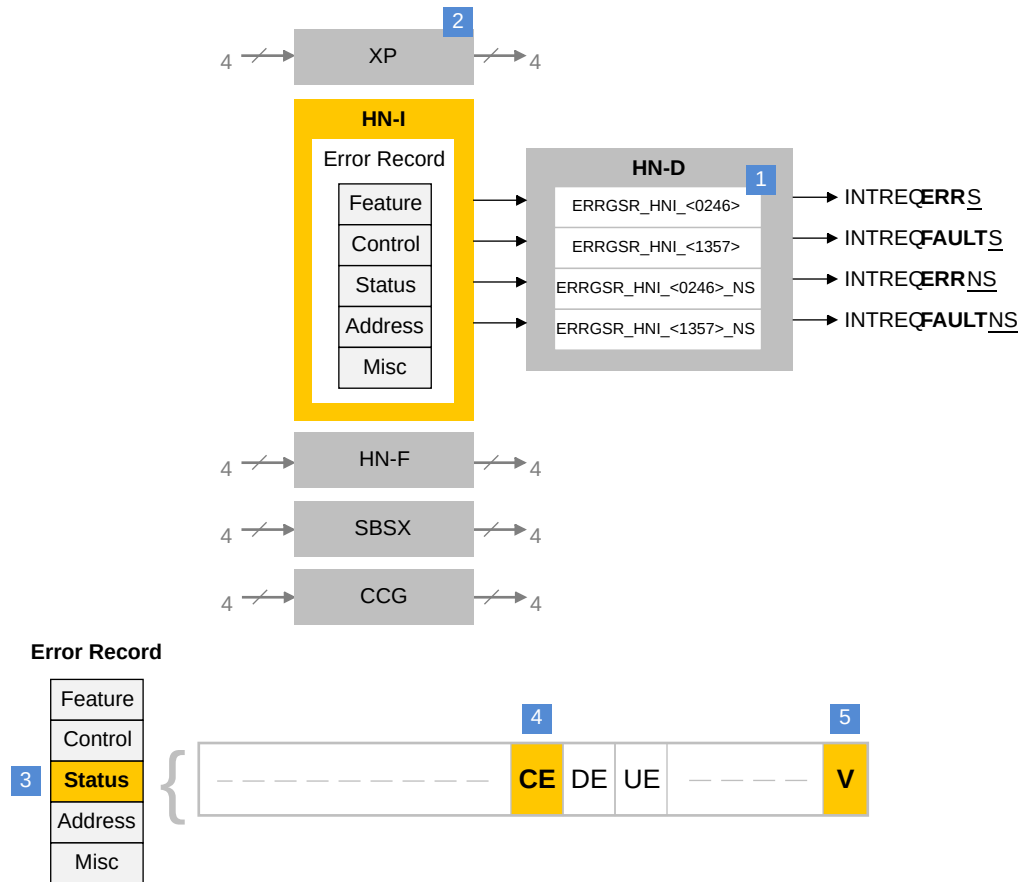
Each device that can detect errors logs the errors in local registers. The device sends error information to the central interrupt handling unit in the HN-D. The HN-D contains four sets of five error groups, which are based on the device type of the error source. The sets consist of a Secure and Non-Secure group for errors, and a Secure and Non-Secure group for fault-type errors. The groups are represented by *Error Group Status Registers* (ERRGSRs).

Error interrupt handler flow example

The following sequence of events and figure describe the process for determining the error source and type of an HN-I generating an interrupt request:

1. The HN-D generates an interrupt for one of the five error group types.
2. The error group indicates the error source device type, which can be:
 - XP
 - HN-I, error source in this example
 - HN-F
 - SBSX
 - CCG
3. The bit location within the error group indicates the logical ID of that device type. In this case, it reveals an HN-I error, the HN-I Error Record Status block for this example.
4. The Status block of the Error Record for the specific XP, HN-I, HN-F, SBSX, or CCG indicates the type of error.
5. The Address and Misc blocks of the Error Record provide further details regarding error root cause, in this case a Corrected Error.
6. The Valid bit is also asserted.
7. To clear the asserted interrupt on the pin, the valid bit of the error status has to be cleared.

Figure 3-27: Error interrupt handler flow example



Each device type has up to 16 ERRGSRs, depending on how many devices of that type are present in the CMN-700 system. For example, the following table shows a possible configuration of the MXP ERRGSRs.

Table 3-22: Example MXP ERRGSR configuration

ERRGSR name	Register offset	Error group
por_cfgm_errgsr_mxp_0	16'h3000	MXP_<63:0> error status
por_cfgm_errgsr_mxp_1	16'h3008	MXP_<63:0> fault status
por_cfgm_errgsr_mxp_2	16'h3010	MXP_<127:64> error status
por_cfgm_errgsr_mxp_3	16'h3018	MXP_<127:64> fault status
por_cfgm_errgsr_mxp_0_NS	16'h3040	MXP_<63:0> error status NS
por_cfgm_errgsr_mxp_1_NS	16'h3048	MXP_<63:0> fault status NS
por_cfgm_errgsr_mxp_2_NS	16'h3050	MXP_<127:64> error status NS
por_cfgm_errgsr_mxp_3_NS	16'h3058	MXP_<127:64> fault status NS

If CMN-700 has ≤64 MXPs, only por_cfgm_errgsr_mxp_0, por_cfgm_errgsr_mxp_1, por_cfgm_errgsr_mxp_0_NS, and por_cfgm_errgsr_mxp_1_NS are present.

Each device that can detect errors has five Error Record registers that contain the error type, along with other information such as the address and opcode. Error types include *Corrected Error* (CE), *Deferred Error* (DE), and *Uncorrected Error* (UE).

3.5.1 Error Detection and Deferred Error values

For XP, the default values of *Error Detection* (ED) and DE depend on build time parameters.

Only the HN-F has CE counters that are implemented in the ERRMISC register. The default values of UI, FI, and CFI are 2'b10, which enables control for the interrupt generation. The following table contains default values of ED and DE for XP.

Table 3-23: Default values of ED and DE for XP

POR_FLIT_PAR_EN	POR_DATACHECK_EN	MXP_DEV_DATACHECK_EN		ED	DE
		P0	P1		
0	0	0	0	2'b00	2'b00
0	1	0	0	2'b01	2'b01
		0	1	2'b01	2'b01
		1	0	2'b01	2'b01
		1	1	2'b00	2'b00
1	0	0	0	2'b01	2'b00
1	1	0	0	2'b01	2'b01
		0	1	2'b01	2'b01
		1	0	2'b01	2'b01
		1	1	2'b01	2'b00

For SBSX, if the `AXDATAPOISON_EN` parameter is not set, the default values of ED and DE are 2'b01. If the parameter is set, the default values are 2'b00.

For HN-I and HN-F, the default values of ED and DE are always 2'b01.

All fields are required, even though only HN-F has CE counters that are implemented in ERRMISC.

The default values of UI, FI, and CFI must be 2'b10, indicating that the interrupt generation is controllable.

3.5.2 HN-F error handling

Errors are reported at the HN-F for various reasons.

Request errors at HN-F

The HN-F detects:

- ECC errors in SF Tag, SLC Tag, and Data RAMs.

- Data check and poison errors on DAT flits.
- *Non-Data Errors* (NDEs) on responses.
- Memory Address Decode Errors

ECC errors in SF Tag, SLC Tag, and Data RAMs

HN-F detects single-bit and double-bit ECC errors in the SF Tag, SLC Tag, and Data RAMs. It can correct single-bit ECC errors. Such errors are logged and reported as CEs.

The source of the double-bit ECC errors determines how they are handled.

SLC Data RAM

- Logged and reported as DEs.
- Propagated to the data consumer in the form of data poison.

SF Tag RAM

- Logged and reported as DEs.
- Not propagated to the requestor.
- The SF Tag RAM in the HN-F is disabled following the first occurrence of the double-bit ECC error.

SLC Tag RAM

- Fatal error.
- Logged and reported as UEs.
- Propagated to the requestor as NDEs in the responses.

Data check and poison errors on DAT flits

If data is allocated by the HN-F, the HN-F detects data check errors and poison error on the data flits. In such cases, HN-F logs and reports the data check error as a DE. If HN-F allocated the data in SLC Data RAM, it converts the data check error into data poison for all subsequent requests to this cache line.

If `cmn_hns_aux_ctl.hnf_poison_intr_en == 1`, then the poison errors originating from HN-F are logged and reported as UEs.

NDEs on responses

HN-F can receive NDEs from other data and response sources such as RN-F, RN-I, and SN-F. If the cache line was allocated in SLC Data RAM, it is logged and reported as a UE. If the cache line is not allocated in HN-F SLC, it propagates the errors to the requestor as an NDE. If SN-F is not CHI-E interface and can send mixed RESPERR in data responses. RN-F and RN-I requires to handle the error properly.

Memory Address Decode Error Handling

HN-F can receive requests and generate SLC and SF victims that HNSAM detects as an invalid memory region. One example is to support CXL hot-plugging memory. Upon detecting any memory

address decode errors, HN-F completes transactions internally without sending any SN requests, and logs decode errors as UE/DE based on configuration register.

HN-F Configuration Registers:

- If `poison_on_mem_addr_dec_err_en` is set, HN-F returns all 0s in data with `DAT.poison` bits set for a read request that has Memory Address Decode Error. Otherwise, HN-F returns all 1s in data with all `DAT.poison` bits cleared
- If `ue_on_mem_addr_dec_err` is set, HN-F logs Memory Address Decode Error as UE. Otherwise, HN-F logs it as DE

3.6 Transaction handling

The handling of certain CHI transaction types and fields differs according to the CMN-700 device type.

Some devices fully support certain transaction types or fields, whereas others do not do any processing of those transactions. Furthermore, some transaction types are unsupported, such as barriers.

3.6.1 Atomics

CMN-700 supports atomic accesses to both cacheable and non-cacheable main memory locations. Atomic accesses to I/O locations are not supported.

3.6.1.1 Atomic requests in HN-F

The HN-F completes all CHI atomic requests that it receives, both for Cacheable and Non-cacheable transactions.

For Cacheable transactions, the HN-F completes any appropriate coherent actions and, if necessary, obtains the targeted cache line from memory. The HN-F then completes the required atomic operation and issues the appropriate response, with or without data.

For Non-cacheable transactions, the HN-F does not send an atomic request to the SN. As the final PoS/PoC for all memory traffic, the HN-F is able to issue a read to the SN, atomically update the copy of the data in the HN-F, and then write back the result to the SN. This approach means that the SN never receives CHI atomic requests, as the HN-F completely handles the requests.

3.6.2 Exclusive accesses

CMN-700 supports exclusive accesses to both Shareable and Non-shareable locations.

See the *AMBA® 5 CHI Architecture Specification*.

3.6.2.1 Exclusive accesses in HN-F

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

RNs generate ReadNoSnp and WriteNoSnp Exclusives for memory locations that are marked Non-cacheable or Device. ReadPreferUnique, MakeReadUnique, ReadShared, ReadClean, ReadNotSharedDirty, and CleanUnique exclusives are used for shareable and coherent memory locations.

Each HN-F in CMN-700 can support tracking of up to 512 logical processors for non-Cacheable exclusive operations:

- In configurations with up to 64 RN-Fs, HN-F supports 64 exclusive monitors.
- In configurations above 144 RN-Fs, the total number of exclusive monitors is equivalent to the total number of local and remote RN-Fs, RN-Is, and RN-Ds, up to a maximum of 512 exclusive monitors.

For non-Cacheable exclusive, the system programmer must ensure that there are no more logical processors capable of concurrently sending exclusive operations than the number of exclusive monitors.

Cacheable exclusive is implemented based on Snoop Filter and not constrained by the number of exclusive monitors.

3.6.3 Completer Busy indication

Transaction completers can use the *Completer Busy* (CBusy) field to indicate their current level of activity. RNs use this indication to determine whether to throttle outgoing traffic.

CMN-700 implements the CBusy indication function in the following node types:

- HN-F
- SBSX
- CCRA

DN, HN-I, HN-P, and HN-T always drive the CBusy values as 0b000.



References to SN-F nodes in this section also apply to SBSX

HN-F CBusy

HN-F uses the *Point-of-Coherency Queue* (POCQ) occupancy level to indicate its current activity level. The following table shows the default CBusy values for a 32-entry POCQ. These values represent the default HN-F CBusy response behavior to RNs.

Table 3-24: HN-F POCQ CBusy thresholds for 32e POCQ

CBusy[2:0]	Tracker occupancy level
0b011	≥24
0b010	≥16
0b001	≥8
0b000	<8

HN-F supports a multisource mode indication in CBusy responses to the RNs. The CBusy[2] bit is used to represent this mode. A CBusy[2]=1 indicates two or more RNs have outstanding requests pending in the HN-F POCQ at the time of response. This mode is enabled by default. HN-F can be programmed to exclude any outstanding RN-I requests in the POCQ when calculating multisource mode. It can be enabled by programming the `cmn_hns_cbusy_limit_ctl.hnf_cbusy_mtbit_exclude_rni` to 1'b1.

HN-F also supports an alternate mode for the CBusy[2] field. It can be programmed to respond with CBusy[2] = 1 when the total number of active requests in the POCQ exceeds a programmable threshold. This mode is in lieu of the multisource mode. The HN-F is programmed to respond busyness in CBusy[1:0] based on read or write request types. The busy indication is only specific to the request category.

The overall occupancy, the total of the read and write request type entries, of POCQ is not available to the RN using CBusy[1:0]. Therefore the above threshold occupancy mode can be used to indicate the complete tracker occupancy and throttle requests accordingly from the RN. Programming `cmn_hns_cbusy_mode_ctl.mt_alt_mode_en` = 1 enables this mode. The threshold for triggering this above threshold indicator can be programmed in `cmn_hns_cbusy_mode_ctl.poc_high_watermark`.

SBSX CBusy

SBSXs only drive CBusy on returning TXDAT flits targeting RNs. These nodes use two hierarchical trackers to drive the CBusy field: ReqTracker and DART. The CBusy field reflects the occupancy levels of both trackers combined. Similar to HN-F, the activity thresholds are programmable. The following table shows the default occupancy threshold for 96 entry trackers.

Table 3-25: SBSX tracker CBusy thresholds

CBusy[2:0]	Tracker occupancy level
0b011	≥72
0b010	≥48
0b001	>24
0b000	<24

SBSXs do not use the multi-source mode bit, so CBusy[2] is always set to 0b0.

CML CBusy

RA uses the request tracker (RHT) occupancy level to indicate the current activity level. The following table shows the default values for a 256 entry RHT. This behavior is the default mode of the RA outgoing CBusy in all responses to RNs.

Table 3-26: RA RHT CBusy thresholds

CBusy[2:0]	Tracker occupancy level
0b011	≥192
0b010	≥128
0b001	≥64
0b000	<64

RA does not use the multi-source mode bit, so CBusy[2] is always set to 0b0.

In addition, in SMP mode, CCG implements passing through CBusy from remote HN-F, HN-I, or HN-P to the requesting RN.

CCRA implements the following software configurable bits to send:

1. CCRA CBusy
2. Remote CBusy coming on data or dataless late completion responses. Intermediate responses, such as DBID, CompDBID, and ReadReceipt, have RA CBusy.
3. Greater of the above two: Applicable to cases where remote CBusy can be sent.

3.6.3.1 Advanced CBusy handling in HN-F

CMN-700 HN-F supports advanced CBusy handling and request throttling to SN-F.

HN-F to RN CBusy handling

The responses that are sent from HN-F to RN through RSP and DAT channels carry CBusy values. HN-F has multiple different modes to determine how the CBusy values are specified in the response messages.

HN-F can be configured to respond to RNs with a CBusy value that reflects one of the following options:

- Total number of outstanding requests in the HN-F POCQ (default mode)
- Independent CBusy values for reads and writes:

CompData type requests (All Read* requests)

CBusy value is based on number of outstanding reads in the POCQ

Comp type requests (Writes, Evict, atomics, CMOs)

CBusy value is based on number outstanding writes in the POCQ

- Return SN-F CBusy value instead of returning value that is based on HN-F POCQ:
 - Read requests receive the Read CBusy of the SN-F
 - Write requests receive the Write CBusy of the SN-F
- Return whichever CBusy value is the highest between HN-F POCQ and SN-F
- MPAM Part ID based CBusy propagation from SN-F

Comp type requests can be further filtered into the following categories:

- CopyBack type requests (Evict, WriteClean, WriteEvictFull, or WriteBack*)
- NonCopyBack type requests (including WriteNoSnP*, WriteUnique*, Combined Write, (P)CMOs, and atomics)

Write filtering of CopyBack versus NonCopyBack types is only supported when you configure HN-F to respond with the CBusy of the POCQ. Write filtering is not supported if the HN-F returns the CBusy value of the SN-F.

The following table shows the format of the `cmn_hns_cbusy_limit_ctl` register. This register controls the HN-F CBusy threshold for Read requests.

Table 3-27: `cmn_hns_cbusy_limit_ctl` register for CBusy thresholds, all requests or read types

Bits	Name	Description
[7:0]	<code>cmn_hns_cbusy_low_limit</code>	Specifies the POC valid threshold at which HN-F is considered least busy
[15:8]	<code>cmn_hns_cbusy_med_limit</code>	Specifies the POC valid threshold at which HN-F is considered medium busy
[23:16]	<code>cmn_hns_cbusy_high_limit</code>	Specifies the POC valid threshold at which HN-F is considered very busy
[48]	<code>cmn_hns_cbusy_rd_wr_types_en</code>	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 <code>sn_cbusy_prop_en</code> = 0b1 to propagate the SN CBusy.
[63]	<code>cmn_hns_cbusy_mtbit_exclude_rni</code>	Allows HN-F to ignore outstanding read requests from RN-I when calculating busyness

The following table shows the format of the `cmn_hns_cbusy_write_limit_ctl` register. This register controls the HN-F CBusy threshold for Write requests.

Table 3-28: Register for CBusy thresholds, write requests

Bitfield	Field	Description
[7:0]	<code>cmn_hns_cbusy_low_limit</code>	Specifies the POC valid threshold at which HN-F is considered least busy
[15:8]	<code>cmn_hns_cbusy_med_limit</code>	Specifies the POC valid threshold at which HN-F is considered medium busy
[23:16]	<code>cmn_hns_cbusy_high_limit</code>	Specifies the POC valid threshold at which HN-F is considered very busy
[48]	<code>cmn_hns_cbusy_sep_copyback_types</code>	When set, HN-F calculates CBusy based on outstanding CopyBack and NonCopyBack type requests independently in the HN-F POCQ

The following table shows the CBusy values that are returned to RNs according to programming.

Table 3-29: HN-F CBusy value propagation according to programming

<code>cmn_hns_adv_cbusy_mode_en</code>	<code>cmn_hns_cbusy_rd_wr_types_en</code>	<code>sn_cbusy_prop_en</code>	<code>cbusy_highest_of_all_en</code>	<code>cbusy_mpam_tbl_en</code>	CBusy value passed to RN
0b0	x	x	x	0b0	POCQ CBusy value is returned
0b1	0b0	x	x	0b0	POCQ CBusy value is returned
0b1	0b1	0b0	0b0	0b0	POCQ CBusy value for read or write is returned, according to the request type
0b1	0b1	0b1	0b0	0b0	SN CBusy value for read or write is returned for the corresponding SN group, according to the request type

cmn_hns_adv_cbusy_mode_en	cmn_hns_cbusy_rd_wr_types_en	sn_cbusy_prop_en	cbusy_highest_of_all_en	cbusy_mpam_tbl_en	CBusy value passed to RN
0b1	0b1	x	0b1	0b0	Highest of either the SN or POCQ CBusy value for read or write is returned, according to the request type
0b1	0b0	x	x	0b1	Reflects the SN CBusy for the corresponding MPAM PartID in both Read and Write response
0b1	0b1	x	x	0b1	Reflects MPAM partID based SN CBusy for Write responses. Read responses receive POCQ occupancy based CBUSY in RSP/DATA flits.

Where applicable, HN-F returns the read or write CBusy value according to opcode type.

Write CBusy values can be further separated into CopyBack and NonCopyBack values using the cmn_hns_cbusy_sep_copyback_types field. This separation only applies when HN-F is programmed to propagate the POCQ CBusy values.

In this mode, CopyBack write type values account for pending WriteClean*, WriteBack*, WriteEvictFull*, and Evict type operations.

NonCopyBack write type values account for all other pending write operations (WriteUnique*, WriteNoSnP*). Combined Write* and (P)CMO operations are counted towards NonCopyBack types. Standalone CMOs are not counted towards either of the CopyBack or NonCopyBack type requests.

HN-F to SN-F CBusy based throttling

HN-F can identify two groups of memory controllers using a configuration bit for each SN. These groups are known as Group A or Group B. You can use the two groups to identify fast and slow memory types. Therefore, the HN-F can handle traffic to and from the two types independently of each other.

HN-F can track the read and write busyness to each SN-F group over a configurable transaction window. It can be programmed to track the last 128 or 256 transactions. When HN-F has received as many responses from SN-F, it measures the current busyness for each group of SN and request types (read and write). The measured busyness is then used to throttle the traffic to SN-F appropriately.

The threshold for measuring the CBusy for the last 128 or 256 transactions is also configurable. For example, consider a scenario where HN-F is programmed to calculate the last 128 CBusy responses. HN-F tracks the number of times it receives CBusy values of 0b00, 0b01, 0b10, and 0b11 for each SN group.

In this example, the HN-F receives more than 16 CBusy = 0b11 responses from Group 0 SN-Fs out of the last 128 responses. In this case, HN-F treats the final SN-F CBusy value as 0b11 for the subsequent 128 transactions while continuing to accumulate new CBusy response values.

HN-F can be configured to throttle outgoing requests in either a static mode or a default dynamic mode:

- Static throttling mode: HN-F controls the outstanding transactions at any point for a given SN group and request to values as programmed in `cmn_hns_cbusy_resp_ctl` register:
 - CBusy = 0b11 (Very busy): HN-F will throttle back outstanding transactions to value programmed in `cbusy_sn_static_ot_count_cbusy11` field.
 - CBusy = 0b10 (Medium busy): HN-F will throttle back outstanding transactions to value programmed in `cbusy_sn_static_ot_count_cbusy10` field.
 - CBusy = 0b01 (Low busy): HN-F will throttle back outstanding transactions to value programmed in `cbusy_sn_static_ot_count_cbusy01` field.
 - CBusy = 0b00 (Not busy): HN-F can issue as many requests as the number of POCQ entries
- Dynamic throttling mode: The number of *Outstanding Transactions* (OTs) can be dynamically throttled according to programmed values. It can be configured to increment or decrement the transaction count by one, two, four, or eight transactions after every 128 or 256 transaction window (as programmed).
 - CBusy = 0b11 (Very busy): Decrement the OT count
 - CBusy = 0b10 (Medium busy): No change to the current OT count
 - CBusy = 0b01 (Low busy): Increment the OT count
 - CBusy = 0b00 (Not busy): Increment the OT count



`cbusy_sn_static_ot_count_cbusyXX` fields must never be programmed to 0 as it can stall forward progress.

Throttling request to SN can be disabled by programming `cbusy_sn_req_throttle_dis` config bit to 0b1. When you configure an HN-F to respond to RNs with the CBusy value of an SN-F, HN-F can propagate the CBusy value according to the SN-F group that the request targets. For example, consider an RN-F sending a read request that is targeting SN group A. The RN can receive the CBusy value for a group A SN, even if the request hits in SLC and therefore the HN-F completes the request.

The following table shows the format of the `cmn_hns_cbusy_resp_ctl` register. This register controls the CBusy responses.

Table 3-30: `cmn_hns_cbusy_resp_ctl` register for configuring CBusy value on responses

Bits	Name	Description: Controls the CBusy responses
[0]	<code>sn_cbusy_prop_en</code>	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 <code>cmn_hns_cbusy_limit_ctl</code> and <code>cmn_hns_cbusy_write_limit_ctl</code> .
[4]	<code>cbusy_highest_of_all_en</code>	When set to 0b1, HN-F responds with the CBusy values from the highest of group A and group B
[7]	<code>cbusy_sn_static_ot_mode_en</code>	Enables the static OT throttling to SN
[21:16]	<code>cbusy_sn_dynamic_ot_count</code>	Count by which the OT count is incremented or decremented for dynamic OT throttling

HN-F throttles only dynamic credit requests to SN-F by default. Requests that were retried and have received a static credit grant from SN-F are allowed to bypass the throttling mechanism. HN-F can be programmed to also throttle static credit requests by setting the `cbusy_sn_retried_req_throttle_en` field to `0b1`.

The following table shows the format of the `cmn_hns_sam_sn_properties1` register. This register controls the group to which each SN belongs.

Table 3-31: Per SN group identifier in `cmn_hns_sam_sn_properties` registers

Field	Description
<code>sn0_group</code>	0b0 Group A 0b1 Group B
<code>sn1_group</code>	
<code>sn2_group</code>	
<code>sn3_group</code>	
<code>sn4_group</code>	
<code>sn5_group</code>	
<code>sn6_group</code>	
<code>sn7_group</code>	
<code>Region0_sn_group</code>	
<code>Region1_sn_group</code>	

The following table shows the format of the `cmn_hns_cbusy_sn_ctl` register. This register controls the CBusy sampling.

Table 3-32: `cmn_hns_cbusy_sn_ctl` register for CBusy sampling control

Bitfield	Field	Description
[9:0]	<code>cmn_hns_cbusy_threshold_cntr01</code>	CBusy threshold at which SN-F is considered busy for Counter_01
[25:16]	<code>cmn_hns_cbusy_threshold_cntr10</code>	CBusy threshold at which SN-F is considered busy for Counter_10
[41:32]	<code>cmn_hns_cbusy_threshold_cntr11</code>	CBusy threshold at which SN-F is considered busy for Counter_11
[56:48]	<code>cmn_hns_cbusy_txn_cnt</code>	Number of SN responses over which the CBusy counters are tracked

HN-F continues to propagate the multi-source bit (`CBusy[2]`) in the advanced modes.

HN-F supports MPAM PartID based CBusy propagation to RN-F. This mode is enabled by programming `mpam_tbl_en=1`. When enabled, HN-F captures the last CBusy value in Read and Write responses from SN for each MPAM PartID. This CBusy value is then propagated to RN-F for corresponding MPAM PartID responses.

For specific use cases where the SN CBusy value must be aggregated in the MPAM Part ID table, HN-F supports an alternate mode of CBusy capture for SN responses. When this mode is enabled using `cmn_hns_cbusy_mode_ctl.cbusy_alt_mode_en=1`, HN-F compacts the 3 bits of SN CBusy into 2-bit CBusy as follows:

- **Default_Mode:** `CBusy[1:0] = SN_CBusy[1:0]`
- **Alt_Mode:** `CBusy[1] = SN's CBusy[2]`, `CBusy[0] = (SN_CBusy[1] & SN_CBusy[0])`

HN-F also supports an alternate mode of CBusy[2] to indicate the POCQ occupancy being higher than a certain threshold. This mode is enabled when `mt_alt_mode_en=1`. The following table shows the `mt_alt_mode` modes. For example, if `mt_alt_mode_en=0`, `CBusy[2]=1` indicates that there are requests from more than one RN active in the POCQ.

Table 3-33: mt_alt_mode modes

Mode	Mode	Description
<code>mt_alt_mode_en</code>	<code>cmn_hns_cbusy_rd_wr_types_en</code>	CBusy[2] MT bit
1'b0	X	Multi source mode: Requests from more than one RN is active in POCQ
1'b1	1'b0	POC occupancy is higher than <code>poc_high_watermark</code>
1'b1	1'b1	Write responses: POC occupancy > <code>poc_high_watermark</code> , Read responses: Requests from more than one RN is active in POCQ

3.6.4 StrongNC RSVDC handling

CMN-700 supports an optional StrongNC subfield in the RSVDC field on CHI REQ channel and AxUser field on AXI interface.

To enable the StrongNC subfield and its associated functionality, set the `RSVDC_STRONGNC_EN` parameter to 1.

StrongNC subfield in the RSVDC field is propagated through RN-I, RN-D, HN-F, HN-I, HN-P, HN-D, HN-T, and CML_SMP blocks. StrongNC subfield is not preserved for traffic that targets SBSX or non-SMP CML links.

For StrongNC requests, HN-F propagates StrongNC subfield in REQ.RSVDC field and sends StrongNC requests directly to SN-F. HN-F does not return data from SLC with SLC hit, nor send out snoops with SF hit. SN-F must be able to process StrongNC requests and return data. StrongNC writes clear exclusive monitors inside HN-F.



RN-F does not support StrongNC requests and always drive the StrongNC bit in the REQ.RSVDC field to 0. RN-I and RN-D support non-cacheable, non-exclusive ReadNoSnp, WriteNoSnpFull, WriteNoSnpPtl with INVALID TagOp for StrongNC requests. If StrongNC requests fall into an HN-F OCM region, HN-F ignores the StrongNC subfield and process the requests like normal OCM requests.

3.7 Quality of Service

CMN-700 includes end-to-end QoS capabilities which support latency and bandwidth requirements for different types of devices.

The QoS device classes are:

Devices with bounded latency requirements

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

Latency-sensitive devices

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

Bandwidth-sensitive devices

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

Bandwidth-hungry devices

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



A device can be classified into one or more of these classes, depending on its workload requirements.

Support for these different types of devices and their resulting traffic is included in the AMBA® 5 CHI protocol and in the entirety of CMN-700 microarchitecture. Each component in CMN-700 contributes to the overall QoS microarchitecture.

3.7.1 Microarchitectural QoS support

The QPV of RN requests must be modulated depending on how well or poorly their respective QoS requirements are met.

3.7.1.1 HN-F QoS support

The HN-F includes the following QoS support mechanisms, if configured to use QoS based classes:

QoS decoding in HN-F

The HN-F interprets the 4-bit QPV at a coarser granularity, as the following table shows. See section [4.5 HN-F class-based resource allocation and arbitration](#) on page 155.



The following table has default configuration and is software-programmable.

Table 3-34: QoS classes in HN-F

QoS value	Class	Dedicated	Contended min	Max allowed
15	Class 0	0	POCQ_ENT / 4	POCQ_ENT - 1
14-12	Class 1	0	POCQ_ENT / 4	POCQ_ENT - 2
11-8	Class 2	0	POCQ_ENT / 4	POCQ_ENT / 2
7-0	Class 3	0	POCQ_ENT / 4	POCQ_ENT / 8

QoS class and POCQ resource availability

The HN-F includes a multi-entry structure, the *Point-of-Coherency Queue* (POCQ), from which all transaction ordering and scheduling is performed. The POCQ buffers are shared resources for all QoS classes, with one entry being reserved for internal use. POCQ is partitioned so that different classes can use configurable number of entries as dedicated, max_allowed or contended_min as shown in the previous figure, ensuring bandwidth and latency requirements of higher priority transactions are met. See [4.5 HN-F class-based resource allocation and arbitration](#) on page 155.

3.8 HN-S

The *Super Home Node* (HN-S) is responsible for managing part of the local address space and managing local coherency for part of the remote address space.



Any descriptions related to the HN-F also apply to the HN-S.

The HN-S consists of the following:

System Level Cache

The *System Level Cache* (SLC) is shared between HBT and LBT lines.



Part of the SLC used for LBT lines is referred as *Local Coherent Cache* (LCC).

The SLC is a last-level cache for HBT lines. The SLC allocation policy is exclusive for data lines, except where sharing patterns are detected and pseudo-inclusive for code lines, as indicated by the RN-Fs. All code lines can be allocated into the SLC on the initial request.

When MTE is enabled, SLC stores data and tags.

For more information about how to partition SLC between HBT and LBT lines, see [3.8.4 HN-S SLC and LCC capacity partitioning](#) on page 110 .

Combined PoS/PoC

The combined *Point-of-Serialization* (PoS) and *Point-of-Coherency* (PoC) is responsible for the ordering of all memory and snoop requests sent to the HN-S. Ordering includes serialization of multiple outstanding requests/snoops and actions to the same line, and request ordering requires the RN-F.

Snoop Filter

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

Each HN-S in the system is configured to manage a specific portion of the overall local address space and remote address space. The entire DRAM space is managed through the combination of all HN-S devices in the system.



Note

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

3.8.1 SAM support for HN-S

The CMN-700 SAM provides a target-type to support HN-S. You must configure every HN-F as an LBT or HBT using a bit vector. For a given address, SAM looks up the HN-F target-ID and provides whether the address is *Home Bound Target* (HBT) or *LCN Bound Target* (LBT).

Table 3-35: Configuration table for HN-S and HN-F target types

Configuration (input)		outputs			
HTG target type	CPA enabled per HNF target	HBT/LBT indication	CML_SMP/CXL indication	rnsam_tgtid_out	rnsam_sn_tgtid_out
HNS tgt type	Local HNF	HBT	-	Local HNS	SNF/CXSA
	Remote HNF (CPA en)	LBT	CML SMP CXL	Local HNS	CCG node
HNF tgt type	Local HNF	HBT	-	Local HNF	SNF/CXSA
	Remote HNF (CPA en)	HBT	-	CCG node	CCG node

Each CPAG is configured as a CML SMP node or CXL node. For a given address, SAM provides whether the address falls under a CML SMP node or a CXL node.

Table 3-36: RNF/RNI mem type based target selection

LBT vs HBT	CXL vs CML SMP	Mem Type	Target
LBT	CXL	-	HN-S (rnsam_tgtid_out)
	CML SMP	Cacheable	
		NC/Dev	CCG (rnsam_sn_tgtid_out)

Figure 3-28: HNS RNSAM

RNSAM (CHIP-0 view)			
HNF targetID table	cpa_en	LCN bound	CPAG
HNS-0	0	0	Local
HNS-1	0	0	Local
HNS-2	0	0	Local
HNS-3	0	0	Local
...	0	0	Local
(HNS-0)HNF-32	1	1	CPAG0
(HNS-1)HNF-33	1	1	CPAG0
(HNS-2)HNF-34	1	1	CPAG0
(HNS-3)HNF-35	1	1	CPAG0
...
(HNS-0)HNF-124	1	1	CPAG2
(HNS-1)HNF-125	1	1	CPAG2
(HNS-2)HNF-126	1	1	CPAG2
(HNS-3)HNF-127	1	1	CPAG2

SAM programming structures:

For local HN-Fs (CPA_EN=0),HNS node-ids are loaded to the target-id table (CPAG = N/A):

- HN-F target-id = HNS node-id

- SN-F target-id = SNF node-id/CXSA node
- HBT indication is sent out

For remote HNFs (CPA_EN = 1), HNS node-ids are loaded to the target-id table. CPAG is configured with the corresponding port aggregation group:

- HN-F target-id = HNS node-id
- SN-F target-id = CCG node id derived from the configured CPAG
- LBT indication is sent out

You must enable CPAG port aggregation for single CCG nodes as well. Direct loading of CCG nodes to the target-id table scenarios are disabled in the presence of HN-S.

3.8.1.1 RNSAM programming for 2P (2 NUMA) HTG and CPA functionality for HNS systems:

The following example details the programming for a 2 NUMA configuration.

2 chip system, RNF = 16 (CAL2), HNF = 16 (CAL2), SNF = 4, CCG = 4

Example 3-11: Programming a 2 NUMA configuration

Figure 3-29: CHIP-0 RNF RNSAM programming for 2P (2 NUMA)

RNSAM (CHIP-0 view) NUMA				
	HNF targetID table	cpa_en	LCN bound	CPAG
SCG-0	HNS-0	0	0	Local
	HNS-1	0	0	Local
	HNS-2	0	0	Local
	...	0	0	Local
	HNS-7	0	0	Local
SCG-1	(HNS-0)HNF-8	1	1	CPAG0
	(HNS-1)HNF-9	1	1	CPAG0
	...	1	1	CPAG0
	(HNS-7)HNF-15	1	1	CPAG0

```

sys_cache_grp_region0.region0_base_addr = <region start addr>
sys_cache_grp_region0.region0_target_type = 3'b101 <HN-S>
sys_cache_grp_region0.region0_valid = 1'b1 <valid>
hashed_tgt_grp_cfg2_region0.region0_end_addr = <region end addr>

sys_cache_group_hn_count.scg0_num_hnf = 8 <number of HNS in CHIP-0 ( 8x2=16 )
CAL2 >
sys_cache_grp_cal_mode_reg.scg0_hnf_cal_mode_en = 1'b1

sys_cache_grp_hn_nodeid_reg[0-1].nodeid#[0-7] = <HNS node ids in CHIP-0>
sys_cache_grp_hn_cpa_en_reg.hash_cpa_en[0-7] = 0 <CHIP-0 HNS are local>
hashed_target_grp_hnf_lcn_bound_cfg_reg0.lcn_bound_en[0-7] = 0 <CHIP-0 HNS are
local and LCN_BOUND should be set to zeroes>

```

Example 3-12: SCG1 - CHIP-1 NUMA

```

sys_cache_grp_region1.region1_base_addr = <region start addr>
sys_cache_grp_region1.region1_target_type = 3'b101 <HN-S>
sys_cache_grp_region1.region1_valid = 1'b1 <valid>
hashed_tgt_grp_cfg2_region1.region1_end_addr = <region end addr>

sys_cache_group_hn_count.scg1_num_hnf = 8 <number of HNS in CHIP-1 ( 8x2=16 )
CAL2 >

```



```
sys_cache_grp_cal_mode_reg.scg1_hnf_cal_mode_en = 1'b1

sys_cache_grp_hn_nodeid_reg[2-3].nodeid_#[8-15] = <LCN node ids in CHIP-0>
sys_cache_grp_hn_cpa_en_reg.hash_cpa_en[8-15] = 8'hFF <CHIP-1 HNS are remote>
hashed_target_grp_hnf_lcn_bound_cfg_reg0.lcn_bound_en[8-15] = 0 <CHIP-1 HNS are
remote, lcn_bound should be set>

sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg1 = 5'h0 <CPAG0>
sys_cache_grp_hn_cpa_grp_reg.enable_multi_cpa_grp_scg1 = 1'b0 <one CPAG for SCG>
```

Example 3-13: CML programming

```
cml_port_aggr_grp0_add_mask.addr_mask = <CPAG addr/axid mask>
cml_port_aggr_grp_reg0.pag_tgtid[0-3] = <CCG target IDs>
cml_port_aggr_ctrl_reg.cpag_port_type0 = 1'b1 <CML SMP port>
cml_port_aggr_ctrl_reg.cpag_axid_hash_en0 = <enable this for RNI/RND>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 4 <number of CCGs>
```

LCN SAM programming in HNS

LCN SAM is configured to direct the LCN bound traffic to remote chips (SCG[2,3,4]). The LCN SAM is not configured for local NUMA.

```
lcn_hashed_tgt_grp_cfg1_region0.htg_region0_base_addr = <base addr>
lcn_hashed_tgt_grp_cfg1_region0.htg_region0_target_type = 3'b000
lcn_hashed_tgt_grp_cfg1_region0.htg_region0_valid = 1'b1
lcn_hashed_tgt_grp_cfg2_region0.region0_end_addr = <end addr>

lcn_hashed_target_group_hn_count_reg0.htg0_num_hn = 8 < 8x2 (CAL2) = 16 >
lcn_hashed_target_grp_cal_mode_reg0.htg0_hn_cal_mode_en = 1'b1

lcn_hashed_target_grp_hnf_cpa_en_reg0.htg_hnf_cpa_en[0-7] = 8'hFF <remote NUMA>
lcn_hashed_target_grp_cpag_perhnf_reg0.htg_cpag_hnf[0-7] = <CPAG0>
```

3.8.1.2 RNSAM programming for 4P (4 NUMA) HTG and CPA functionality for HNS systems:

The following example details the programming for a 4 NUMA configuration:

4 chip system, RNF = 16 (CAL2), HNF = 16 (CAL2), SNF = 4, CCG = 2 (per chip)

Figure 3-30: CHIP-0 RNF RNSAM programming for 2P (2 NUMA)

RNSAM (CHIP-0 view) NUMA					
	HNF targetID table	cpa_en	LCN bound	CPAG	
SCG-0 (local)	HNS-0	0	0	Local	
	HNS-1	0	0	Local	
	HNS-2	0	0	Local	
	...	0	0	Local	
	HNS-7	0	0	Local	
SCG-1 (chip-1)	(HNS-0)HNF-8	1	1	CPAG0	
	(HNS-1)HNF-9	1	1	CPAG0	
	...	1	1	CPAG0	
	(HNS-7)HNF-15	1	1	CPAG0	
SCG-2 (chip-2)	(HNS-0)HNF-16	1	1	CPAG1	
	(HNS-1)HNF-17	1	1	CPAG1	
	...	1	1	CPAG1	
	(HNS-7)HNF-23	1	1	CPAG1	
SCG-3 (chip-3)	(HNS-0)HNF-24	1	1	CPAG2	
	(HNS-1)HNF-25	1	1	CPAG2	
	...	1	1	CPAG2	
	(HNS-7)HNF-31	1	1	CPAG2	

Example 3-14: SCG - CHIP-0 NUMA

```

sys_cache_grp_region0.region0_base_addr    = <region start addr>
sys_cache_grp_region0.region0_target_type  = 3'b101 <HN-S>
sys_cache_grp_region0.region0_valid        = 1'b1  <valid>
hashed_tgt_grp_cfg2_region0.region0_end_addr = <region end addr>

sys_cache_group_hn_count.scg0_num_hnf     = 8 <number of HNS in CHIP-0 ( 8x2=16 )
CAL2 >
sys_cache_grp_cal_mode_reg.scg0_hnf_cal_mode_en = 1'b1

sys_cache_grp_hn_nodeid_reg[0-1].nodeid_# [0-7] = <HNS node ids in CHIP-0>
sys_cache_grp_hn_cpa_en_reg.hash_cpa_en[0-7] = 0 <CHIP-0 HNS are local >

```

```
hashed_target_grp_hnf_lcn_bound_cfg_reg0.lcn_bound_en[0-7] = 0 <CHIP-0 HNS are
local and LCN_BOUND should be set to zeroes>
```

Example 3-15: SCG - CHIP-1 NUMA

```
sys_cache_grp_region1.region1_base_addr      = <region start addr>
sys_cache_grp_region1.region1_target_type    = 3'b101 <HN-S>
sys_cache_grp_region1.region1_valid          = 1'b1 <valid>
hashed_tgt_grp_cfg2_region1.region1_end_addr = <region end addr>

sys_cache_group_hn_count.scg1_num_hnf       = 8 <number of HNS in CHIP-0 ( 8x2=16 )
CAL2 >
sys_cache_grp_cal_mode_reg.scg1_hnf_cal_mode_en = 1'b1

sys_cache_grp_hn_nodeid_reg[2-3].nodeid_#[8-15] = <LCN node ids in CHIP-0>
sys_cache_grp_hn_cpa_en_reg.hash_cpa_en[8-15] = 8'hFF <CHIP-1 HNS are remote>
hashed_target_grp_hnf_lcn_bound_cfg_reg0.lcn_bound_en[8-15] = 8'hFF <CHIP-1 HNS
are remote, lcn_bound should be set>

sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg1     = 5'h0 <CPAG0>
sys_cache_grp_hn_cpa_grp_reg.enable_multi_cpa_grp_scg1 = 1'b0 <one CPAG for SCG>
```

Example 3-16: SCG - CHIP-2 NUMA

```
sys_cache_grp_region2.region2_base_addr      = <region start addr>
sys_cache_grp_region2.region2_target_type    = 3'b101 <HN-S>
sys_cache_grp_region2.region2_valid          = 1'b1 <valid>
hashed_tgt_grp_cfg2_region2.region2_end_addr = <region end addr>

sys_cache_group_hn_count.scg2_num_hnf       = 8 <number of HNS in CHIP-1 ( 8x2=16 )
CAL2 >
sys_cache_grp_cal_mode_reg.scg2_hnf_cal_mode_en = 1'b1

sys_cache_grp_hn_nodeid_reg[4-5].nodeid_#[16-23] = <LCN node ids in CHIP-0>
sys_cache_grp_hn_cpa_en_reg.hash_cpa_en[16-23] = 8'hFF <CHIP-2 HNS are remote>
hashed_target_grp_hnf_lcn_bound_cfg_reg0.lcn_bound_en[16-23] = 8'hFF <CHIP-2 HNS
are remote, lcn_bound should be set>

sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg2     = 5'h1 <CPAG1>
sys_cache_grp_hn_cpa_grp_reg.enable_multi_cpa_grp_scg2 = 1'b0 <one CPAG for SCG>
```

Example 3-17: SCG - CHIP-3 NUMA

```
sys_cache_grp_region3.region3_base_addr      = <region start addr>
sys_cache_grp_region3.region3_target_type    = 3'b101 <HN-S>
sys_cache_grp_region3.region3_valid          = 1'b1 <valid>
hashed_tgt_grp_cfg2_region3.region3_end_addr = <region end addr>

sys_cache_group_hn_count.scg3_num_hnf       = 8 <number of HNS in CHIP-1 ( 8x2=16 )
CAL2 >
sys_cache_grp_cal_mode_reg.scg3_hnf_cal_mode_en = 1'b1

sys_cache_grp_hn_nodeid_reg[6-7].nodeid_#[24-31] = <LCN node ids in CHIP-0>
sys_cache_grp_hn_cpa_en_reg.hash_cpa_en[24-31] = 8'hFF <CHIP-3 HNS are remote>
hashed_target_grp_hnf_lcn_bound_cfg_reg0.lcn_bound_en[24-31] = 8'hFF <CHIP-3 HNS
are remote, lcn_bound should be set>

sys_cache_grp_hn_cpa_grp_reg.cpa_grp_scg3     = 5'h2 <CPAG2>
```

```
sys_cache_grp_hn_cpa_grp_reg.enable_multi_cpa_grp_scg3 = 1'b0 <one CPAG for SCG>
```

Example 3-18: CPAG0 for SCG1

```
cml_port_aggr_grp0_add_mask.addr_mask = <CPAG addr/axid mask>
cml_port_aggr_grp_reg0.pag_tgtid[0-1] = <CCG target IDs>
cml_port_aggr_ctrl_reg.cpag_port_type0 = 1'b1 <CML SMP port>
cml_port_aggr_ctrl_reg.cpag_axid_hash_en0 = <enable this for RNI/RND>
cml_port_aggr_ctrl_reg.cpag_valid0 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag0 = 2 <number of CCGs>
```

Example 3-19: CPAG1 for SCG2

```
cml_port_aggr_grp1_add_mask.addr_mask = <CPAG addr/axid mask>
cml_port_aggr_grp_reg0.pag_tgtid[2-3] = <CCG target IDs>
cml_port_aggr_ctrl_reg.cpag_port_type1 = 1'b1 <CML SMP port>
cml_port_aggr_ctrl_reg.cpag_axid_hash_en1 = <enable this for RNI/RND>
cml_port_aggr_ctrl_reg.cpag_valid1 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag1 = 2 <number of CCGs>
```

Example 3-20: CPAG2 for SCG3

```
cml_port_aggr_grp2_add_mask.addr_mask = <CPAG addr/axid mask>
cml_port_aggr_grp_reg[0-1].pag_tgtid[4-5] = <CCG target IDs>
cml_port_aggr_ctrl_reg.cpag_port_type2 = 1'b1 <CML SMP port>
cml_port_aggr_ctrl_reg.cpag_axid_hash_en2 = <enable this for RNI/RND>
cml_port_aggr_ctrl_reg.cpag_valid2 = 1'b1
cml_port_aggr_ctrl_reg.num_cxg_pag2 = 2 <number of CCGs>
```

LCN SAM programming in HNS

LCN SAM is configured to direct the LCN bound traffic to remote chips (SCG[2,3,4]). The LCN SAM is not configured for local NUMA.

```
lcn_hashed_tgt_grp_cfg1_region0.htg_region0_base_addr = <CHIP-1 base addr>
lcn_hashed_tgt_grp_cfg1_region0.htg_region0_target_type = 3'b000
lcn_hashed_tgt_grp_cfg1_region0.htg_region0_valid = 1'b1
lcn_hashed_tgt_grp_cfg2_region0.region0_end_addr = <CHIP-3 end addr>

lcn_hashed_target_group_hn_count_reg0.htg0_num_hn = 8 < 8x2(CAL2) = 16 >
lcn_hashed_target_grp_cal_mode_reg0.htg0_hn_cal_mode_en = 1'b1

lcn_hashed_target_grp_hnf_cpa_en_reg0.htg_hnf_cpa_en[0-7] = 8'hFF <remote NUMA>
lcn_hashed_target_grp_cpag_perhnf_reg0.htg_cpag_hnf[0-7] = <CPAG0>
```

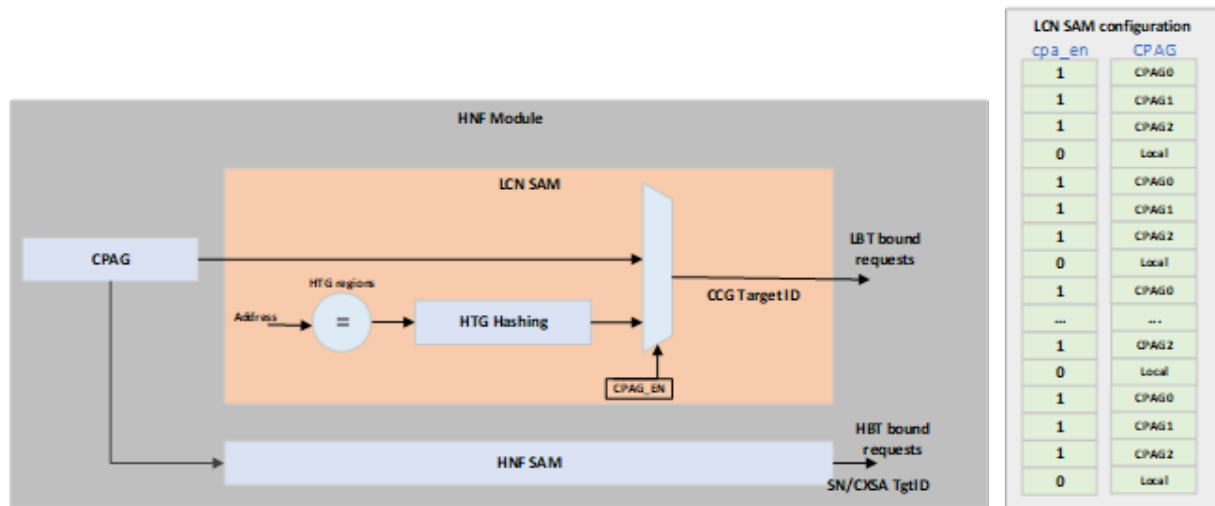
3.8.2 LCN SAM

The LCN SAM is instantiated in HNS to determine the CCG node-ids for the LBT bound transactions and is derived from the RNSAM with only HTG structures and config space that is defined in HN-S.

The LCN SAM HTG programming is similar to the RNSAM HTG structures. CPAG structures are reused from the snoop transactions CCG lookup logic.

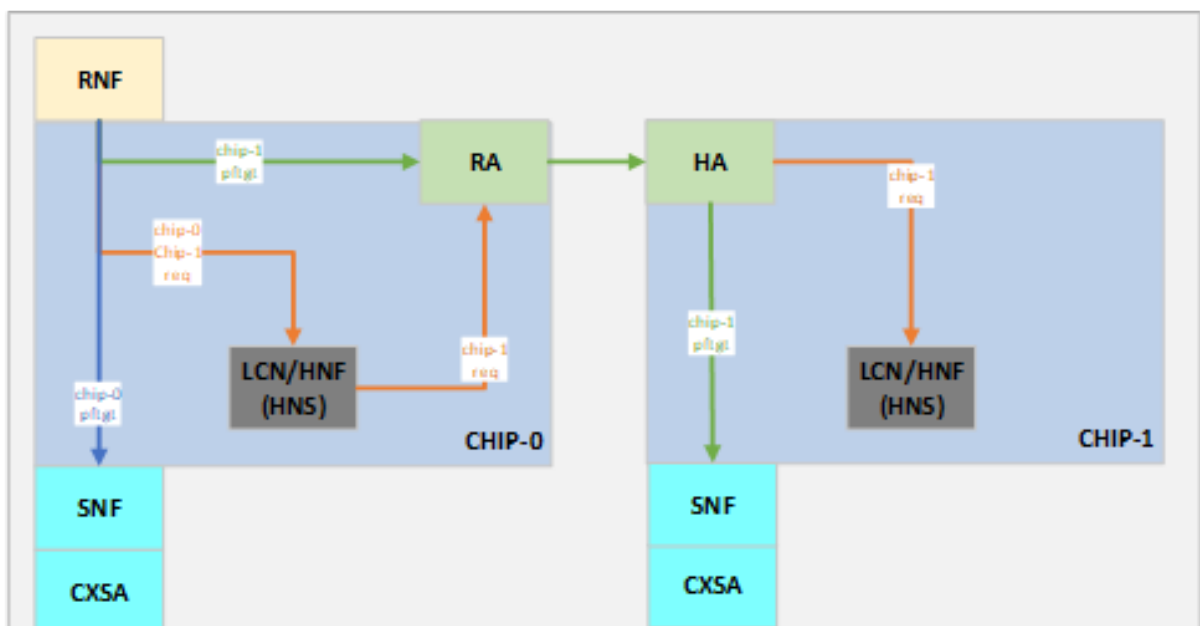
The LCN SAM programming does not require programming HN-F target-ids as LBT bound transactions only direct towards CCG nodes. The LCN SAM requires CPA_EN bit vector and CPA group-ID programming for each HN-F entry.

Figure 3-31: LCN SAM configurations



The following figure shows CHI Request/prefetch transaction target-ID look up from RN SAM and LCN SAM

Figure 3-32: CHI Request/prefetch transaction target-id look up from RN SAM and LCN SAM



3.8.3 HN-S Completer Busy Indication

CMN-700 adds support for Cbusy for LBT addresses. HN-S has multiple different modes to determine how the CBusy values are specified in the response messages for LBT transactions.

HN-S can be configured to respond to RNs with a CBusy value that reflect one of the following options:

1. Total number of outstanding requests in POCQ (Default)
2. Uses the Cbusy responded on late completion responses (Compdata/Comp)

For more information about supported configurations, see [5.2.4.110 cmn_hns_lbt_cbusy_ctl](#) on page 482.

3.8.4 HN-S SLC and LCC capacity partitioning

HN-S supports capacity-based SLC and LCC partitioning, which defines the percentage of cache that LBT and HBT requests can use. SLC and LCC capacity partitioning has following features:

- hns_slc_cmax_allowed and hns_lcc_cmax_allowed registers are set to 7-bit wide. Providing a granularity of 0.78% ($1/2^7$) for SLC and LCC partitioning.
- In HAM mode cache capacity is adjusted for half the cache
- CMN-700 supports address-based locking, including OCM, with SLC and LCC capacity partitioning. Locked ways are accounted for SLC and LCC capacity partitioning.
- SLC and LCC capacity partitioning counters are not accurate when exiting retention state and can result in underflow conditions

CMN-700 supports SLC and LCC capacity partitioning with both eLRU and LSFR cache replacement policies.

CMN-700 supports SLC and LCC capacity partitioning with Source-based SLC cache partitioning and Way-based SLC cache partitioning, but not MPAM. See [4.2.9 Source-based SLC cache partitioning](#) on page 131 and [4.2.10 Way-based SLC cache partitioning](#) on page 132



If hnf_ocm_allways_en bit is set to 1, SLC is fully reserved for *Home Bound Addresses* (HBA), and no *Local Bound Addresses* (LBA) can be allocated into cache.

3.8.5 Configuring non-clustered RN-F tracking in HN-S SF

For HN-S device, `POR_NUM_REMOTE_RNF_PARAM` defines the number of remote chips in a hierarchical system.

To enable non-clustered mode in HNS, set `SF_MAX_RNF_PER_CLUSTER` parameter to 1, and set `POR_NUM_REMOTE_RNF_PARAM` to number of remote chips. The LDID of each Local RN-F and each remote chip is assigned to a single index entry in the SF RN-F vector index.

In a CML configuration with HN-S, all remote chips must have LDIDs assigned to them after the local RN-F LDIDs have been assigned.

For example, consider a 4-chip system with 64 RN-Fs, divided into 16 local RN-Fs on each chip. In this system, LDIDs 0-15 are preassigned to the local RN-Fs. `POR_NUM_REMOTE_RNF_PARAM` is set to 3, and LDIDs 16-18 must then be assigned to 3 remote chips.

For more information about SF tracking in HN-F, see:

- [4.2.11 RN-F tracking in the SF](#) on page 134
- [4.2.12 Non-clustered and clustered modes for SF RN-F tracking](#) on page 135
- [4.2.13 Configuring non-clustered RN-F tracking in HN-F SF](#) on page 135

3.8.6 HN-S error handling

CMN-700 error handling in the HN-F also applies to error handling for the HN-S.

The following are additional rules that only apply to the HN-S:

NDEs on Snoop request

When the HN-S is processing an incoming Snoop request from CCG RA, HN-S can have double-bit ECC error on LCC Tag RAM or NDEs in snoop responses from RN-F. Both are logged and reported as UE, and HN-S doesn't propagate NDE to CCG RA.

NDEs on responses

The HN-S can receive NDEs from other data and response sources such as RN-F, RN-I, and SN-F. For LBT:

- If HN-S sends early responses back to the requester, NDE from remote HN-S is logged and reported as a UE
- If HN-S sends late responses back to the requester, it propagates the errors to the requestor as an NDE

For more information about error handling in the HN-S, see [3.5.2 HN-F error handling](#) on page 88

3.8.7 HN-S POCQ Resource Allocation

CMN-700 has 4 entries reserved in POCQ as follows:

- 2 entries are reserved for SEQ
 - The 1st entry is reserved for SEQ HBT addresses
 - The 2nd entry is reserved for the HBT or LBT
- 1 entry is reserved for incoming snoops
- 1 entry is reserved for HBT transactions from remote chip (CCG-HA)



The total number of dedicated entries programmed in `cmn_hns_pocq_alloc_class_dedicated`, including the reserved entries mentioned above, must not exceed the number of POCQ entries.

For more information HBT POCQ Resource Allocation and on various POCQ resource register programming, see [4.5.2 POCQ resource allocation](#) on page 156

3.8.8 HN-S restrictions

The following describes the restrictions that are present when using an HN-S device.

The HN-S does not support the following features:

- Clustered mode for SF tracking

The following features are supported in HN-S devices with HBT bound traffic only, which targets local memory regions:

- MTE
- RSVDC_PBHA
- RSVDC_METADATA
- StrongNC
- Address-based flushing
- SLC replacement hint



In multi-chip configurations with HNS devices, remote traffic is expected to go through HNS for local coherency resolution. see [3.8.1 SAM support for HN-S](#) on page 101

3.9 HN-F performance events

The HN-F performance analysis counters are used to monitor cache behavior.

For a particular cache, the cache miss rate or hit rate is used to measure the capacity of the cache, and the location for certain applications. To measure the cache miss rate, the performance monitor counters count the number of instances of cache accesses and cache misses.

3.9.1 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 cache performance events are as follows.

Cache miss rate

The cache events that are required to calculate the cache miss rate are:

PMU_HN_CACHE_MISS_EVENT

Counts the total cache misses. A miss results from a first-time lookup and is high priority.

PMU_HNSLC_SF_CACHE_ACCESS_EVENT

The total number of cache accesses. An access is first-time and high priority.



The performance counter architecture allows up to four HNs to collect the cache miss rate for each DTC domain. In a system with multiple DTC domains, more than four HNs can collect the cache miss rate. However, because of the CMN-700 microarchitecture, the cache miss rate that is measured at one HN-F within an SCG is a good proxy for the cache miss rate of the remaining HN-Fs.

Calculate the cache miss rate as follows:

Figure 3-33: Cache miss rate

$$\text{Cache miss rate (\%)} = \frac{\text{Total cache misses}}{\text{Total cache accesses}} \times 100$$

Certain request types can cause multiple cache accesses:

- Lookup
- Tag update
- Victim selection
- Cache fill

Event counting is therefore limited to first time accesses only. For example, for a ReadUnique transaction that leads to an SLC hit, PMU_HNSLC_SF_CACHE_ACCESS_EVENT is only counted the first-time cache lookup is performed. The tag update is not counted as a cache access. Similarly, for Write-Back or Write*Unique transactions with an SLC allocate hint, only the first instance of an SLC lookup is counted as an access and hit or miss. The eventual victim selection and cache fill are not counted as further accesses.

Cache allocations

The cache allocation event counts the number of times an HN-F SLC cache is allocated. It provides an approximate cache usage for this particular application over a specific time slice. This event does not check whether the application has any hot sets.

PMU_HN_CACHE_FILL_EVENT

Counts all cache line allocations to SLC cache.

All cache line writes, that is, Write*Unique, Write-Back, and evictions that are allocated in SLC cache, are counted towards this event.

3.9.2 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 3-34: HN-F message retry rate

$$\text{HN-F message retry rate (\%)} = \frac{\text{HN-F total messages retried}}{\text{HN-F total messages received}} \times 100$$

3.9.3 SF events

There are three snoop event types that can be counted.

The SF performance event types are as follows.

SF miss rate

This event can be used to measure the Snoop Filter efficiency.

PMU_HN_SF_HIT_EVENT

Measures the number of SF hits.

Calculate the SF hit rate as follows:

Figure 3-35: 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 3-36: 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.

3.9.4 System-wide events

The memory controller request retries determine whether the memory controller is the bottleneck in the system, which can cause higher request latencies.

The following events can be counted:

PMU_HN_MC_RETRIES_EVENT

Number of requests that are retried to the memory controller.

PMU_HN_MC_REQS_EVENT

Total number of requests that are sent to the memory controller.

Calculate the retry rate for requests to the memory controller as follows:

Figure 3-37: MC message retry rate

$$\text{MC message retry rate (\%)} = \frac{\text{MC total messages retried}}{\text{MC total messages received}} \times 100$$

3.9.5 Snoop events related to SF clustering

Certain HN-F PMU events can be used to understand the performance impact of SF clustering.

The following events can be counted:

PMU_HN_SNP_SENT_CLUSTER_EVENT

Counts the number of snoops that are sent at the level of a whole cluster. This event does not count individual snoops within a cluster. For example, in a cluster with four RN-Fs, if HN-F sends four snoops, since all four snoops are sent to the same cluster they are counted as one.

PMU_HN_SF_IMPRECISE_EVICT_EVENT

Counts the number of times an evict operation from an RN does not clear the SF tracking because the line was in shared state (imprecise). If there is a single RN-F in the cluster, then the evict operation always clears the tracking for that RN-F. However, in SF clusters, the evict operation must not clear the SF tracking as other RN-Fs might still be accessing this line.

PMU_HN_SF_EVICT_SHARED_LINE_EVENT

Counts the number of times an SF eviction happened to a cache line that was in shared state. This event can be helpful in understanding of the impact of SF pollution in clustered mode.

3.9.6 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.

3.9.7 HN-F PMU event summary

The following table shows a summary of the HN-F PMU events.

Table 3-37: 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

Number	Name	Description
21	PMU_HN_TXRSP_STALL_EVENT	Counts number of times HN-F has a pending TXRSP flit but no credits to upload
22	PMU_HN_SEQ_FULL_EVENT	Counts number of times requests are replayed in SLC pipe because of SEQ being full
23	PMU_HN_SEQ_HIT_EVENT	Counts number of times a request in SLC hit a pending SF eviction in SEQ
24	PMU_HN_SNP_SENT_EVENT	Counts number of snoops sent including directed, multicast, and SF back invalidation
25	PMU_HN_SFBI_DIR_SNP_SENT_EVENT	Counts number of times directed snoops were sent because of SF back invalidation
26	PMU_HN_SFBI_BRD_SNP_SENT_EVENT	Counts number of times multicast snoops were sent because of SF back invalidation.
27	PMU_HN_SNP_SENT_UNTRK_EVENT	Counts number of times snoops were sent because of untracked RN-Fs
28	PMU_HN_INTV_DIRTY_EVENT	Counts number of times SF back invalidation resulted in dirty line intervention from the RN
29	PMU_HN_STASH_SNP_SENT_EVENT	Counts number of times stash snoops were sent
30	PMU_HN_STASH_DATA_PULL_EVENT	Counts number of times stash snoops resulted in data pull from the RN
31	PMU_HN_SNP_FWDDED_EVENT	Counts number of times data forward snoops were sent
32	PMU_HN_ATOMIC_FWD_EVENT	Counts number of time atomic data was forwarded between POC entries
33	PMU_HN_MPAM_REQ_OVER_HARDLIM_EVENT	Counts number of times write request cannot allocate SLC because of being over hard limit
34	PMU_HN_MPAM_REQ_OVER_SOFTLIM_EVENT	Counts number of times write request is above soft limit
35	PMU_HN_SNP_SENT_CLUSTER_EVENT	Counts number of snoops sent to clusters excluding individual snoops within a cluster
36	PMU_HN_SF_IMPRECISE_EVICT_EVENT	Counts number of times an evict operation was dropped because of SF clustering
37	PMU_HN_SF_EVICT_SHARED_LINE_EVENT	Counts number of times a shared line was evicted from SF
38	PMU_HN_POCQ_CLASS_OCCUPANCY_EVENT	Counts the given POCQ occupancy for a given class in HN-F. Class occupancy filtering is programmed in pmu_class_occup_id.
39	PMU_HN_POCQ_CLASS_RETRY_EVENT	Counts number of retried requests for a given class. Class filtering is programmed in pmu_class_occup_id.
40	PMU_HN_CLASS_MC_REQS_EVENT	Counts number of requests sent to MC for a given class. Class filtering is programmed in pmu_class_occup_id.
41	PMU_HN_CLASS_PCRDGNT_BELOW_CONDMIN_EVENT	Counts number of protocol credit grants for a given class when it is above dedicated and below conditional min. Class filtering is programmed in pmu_class_occup_id.
42	PMU_HN_NUM_SN_CBUSY_THROTTLE_EVENT	Counts number of times request to SN was throttled because of CBusy. Event filtering is programmed in pmu_cbusy_snthrottle_sel.
43	PMU_HN_NUM_SN_CBUSY_THROTTLE_MIN_EVENT	Counts number of times request to SN was throttled to the minimum allowed value of 4 because of CBusy. Event filtering is programmed in pmu_cbusy_snthrottle_sel.
44	PMU_HN_SF_PRECISE_TO_IMPRECISE_EVENT	Counts when number of sharers exceeds how many RNs could be precisely tracked in SF

Number	Name	Description
45	PMU_HN_SNP_INTV_CLN_EVENT	Counts the number of times clean data intervened for a snoop request
46	PMU_HN_NC_EXCL_EVENT	Counts the number of times non-cacheable exclusive request arrived at HN-F
47	PMU_HN_EXCL_MON_OVFL_EVENT	counts the number of times exclusive monitor overflowed

4. SLC memory system

This chapter describes the optional SLC memory system which is implemented by HN-Fs in the mesh.

4.1 About the SLC memory system

The SLC memory system consists of the HN-F protocol nodes in CMN-700.

There is a configurable number of instances (1-128) of the HN-F. Each HN-F node or slice has the following features:

- 0KB, 128KB, 256KB, 384KB, 512KB, 1MB, 1.5MB, 2MB, 3MB, or 4MB of SLC Data RAM and Tag RAM
- Combined *Point-of-Coherency* (PoC) and *Point-of-Serialization* (PoS)
- SF size based on SF_NUM_WAYS:
 - SF_NUM_WAYS = 16: 512KB, 1MB, 2MB, 4MB, 8MB
 - SF_NUM_WAYS = 28: 896KB, 1.8MB, 3.6MB, 7.2MB, 14.3MB
 - SF_NUM_WAYS = 32: 1MB, 2MB, 4MB, 8MB, 16MB
- SF Tag RAM configurable to have 16, 28 or 32 ways

Each HN-F in CMN-700 is configured to manage a specific portion of the total address space. For each portion of the address, each HN-F:

- Can cache data in SLC
- Manages PoC and PoS functionality for ordering and coherency
- Tracks RN-F caching in the SF

The SLC memory system has the following features:

- *Physically Indexed and Physically Tagged* (PIPT)
- Coherency granule is a fixed length of 64B. SLC line size is a fixed length of 64B.
- Both SLC 16-way set-associative for 512KB, 1MB, 2MB, 4MB and 8MB sizes. 12-way for 3MB SLC configurations.
- Optionally, CMN-700 supports an *enhanced LRU* (eLRU) cache replacement policy that you can enable by setting a bit in the configuration register. eLRU is Dynamic Biased Replacement Policy. 2 bits per set, or way, track, and predict how soon a cache line is expected to be used again. This information is dynamically adjusted based on a few reference sets. By default, the SLC and SF victim selection policy is:
 - Pseudo random if all ways are valid
 - If there is an invalid way, it is not necessary to select a victim
 - Victim selection is required only if all ways are taken

- SLC and SF arrays:
 - Support one-cycle, two-cycle, or three-cycle non-pipelined tag array
 - Support two-cycle or three-cycle non-pipelined data array
 - SLC Tag, SF Tag, and SLC Data arrays are single-ported, supporting one read or write access with no concurrency available
 - SLC Tag, SF Tag, and SLC Data arrays are ECC SECDED protected, with inline ECC checking and correction
- 16, 24, 32, 48, 64, 80, 96, or 128-entry Address and data buffer, known as the *PoC Queue* (POCQ), to service:
 - All transactions from the CHI interface
 - SLC evictions to the memory controller
 - SF evictions and associated Write-Backs to the memory controller
- Fully configurable clustering of RN-Fs to support up to 512 caching agents tracked in SF
- CMO propagation to SN-F or SBSX:
 - Implements improved PCMO flow that is introduced in CHI-D
 - Conditional CMO propagation to the memory controller to support external DRAM caches
 - HN-F must be explicitly programmed, using the `cmn_hns_sam_sn_properties` register of the HN-F SAM, to enable this propagation to each SN-F
- Protocol flow control using programmable classes:
 - POCQ resources are allocated or rejected for protocol retry according to the class
 - POCQ resources are watermarked for different classes with user-configurable options
 - Class-based static grantee selection for CHI architecture credit return

See [4.5 HN-F class-based resource allocation and arbitration](#) on page 155

- Class-based request selection to memory controller
- Allocation in the SLC from snoop intervention. This feature enables data sharing through the SLC for multiple sharers.
- SLC state includes a caching LDID to detect dynamic read sharing
- Configurable 34-bit, 44-bit, 48-bit, or 52-bit *Physical Address* (PA) support
- PoC and PoS for all snoopable and non-snoopable, and cacheable and Non-cacheable transaction address space
- Supports ECC scrubbing for single-bit ECC errors on SF, Tag RAMs, SLC Tag RAMs, and SLC Data RAMs.
- Software-controlled error injection support to enable testing of software error handler routine
- Power management states to support:
 - Full powerdown of the SLC and SF, with HN-F only mode when both SLC and SF are powered down
 - Half the SLC ways powered down

- Retention for SLC and SF
- SLC full powerdown with SF on, when in SFONLY mode
- Arm *Realm Management Extension* (RME) support in SLC and SF
- Software-configurable (one, two, four, eight, or 12 ways) memory region locking support in the SLC
- Software-configurable (one, two, four, eight, or 12 ways) OCM support in the SLC
 - OCM memory does not require any physical memory backing
- CHI enhancements for:
 - *Direct Cache Transfer* (DCT)
 - *Direct Memory Transfer* (DMT), for an SBSX with 128 bits must not support read interleave.
 - Cache stashing
 - Atomics support
 - Data poison
 - Data parity (data check)
 - Trace tag
- Invisible SLC support:
 - CMN-700 HN-F implements an invisible cache. All accesses (cacheable, non-cacheable, and device types) are checked against the SLC and SF. The SLC cannot be cleaned and invalidated (flushed) by software using Arm architecture set, or way, operations. Software specific to CMN-700 is required to flush the SLC, as this manual describes. Invisible SLC support eliminates the requirement to perform SLC flushes for software context switches from cacheable to non-cacheable.
- HNSAM supports:
 - Up to 64 non-hashed memory regions
 - Up to 8 hashed memory region (supporting two, four, or eight SN-F address hashing and SA aggregated hash functions)
 - Default regions, which support one, three, five, or six SN-F address hashing
- Supports CHI-F MPAM
- Supports CHI-E *Memory Tagging Extensions* (MTE)

4.2 SLC memory system components and configuration

CMN-700 *System Level Cache* (SLC) is a distributed, mostly exclusive, last-level cache that is implemented within the HN-F node.

When a sharing pattern is detected between RN-F clusters, the SLC is optimized to eliminate redundancy for private data lines from the RN-F. The SLC also enables redundancy, or pseudo-inclusion.

The SLC acts as DRAM cache for I/O coherent agents, that is, RN-Is. The SLC enables RN-Is to allocate or not allocate, according to the usage model.

The SF works with the SLC to track coherent lines that are present in the RN-F caches. The SF is fully inclusive of all the lines present in the RN-F caches. SF eviction invalidates the lines from RN-F caches to maintain this inclusion.

Usually, a particular coherent cache line is present only in the system-level cache or SF except when the line is shared between RN-F clusters. In the shared case, the line can be present in both the SLC and the SF.

4.2.1 HN-F configurable options

You can configure the HN-F in several ways.

The HN-F has the following configurable parameters:

- SF size based on SF_NUM_WAYS
 - SF_NUM_WAYS = 16: 512KB, 1MB, 2MB, 4MB, 8MB
 - SF_NUM_WAYS = 32: 1MB, 2MB, 4MB, 8MB, 16MB
- SF 16, 20, 24, 28, or 32 way set associate
- 16, 24, 32, 48, 64, 80, 96, or 128 POCQ entries. >64 configurations may have frequency implications
- One-cycle, two-cycle, or three-cycle Tag RAM arrays. For a given configuration, both SLC Tag and SF Tag have the same latency.
- Two-cycle or three-cycle Data RAMs, data, and SF array RAMs. All Data RAMs have the same latency.
- RN-F clustered mode and number of RN-Fs that are in each cluster using SF_MAX_RNF_PER_CLUSTER. SF_MAX_RNF_PER_CLUSTER supports values of 1, 2, 4, and 8.
- Number of extra bits in the SF RN-F tracking vector to allow for hybrid clustering of RNs using SF_RN_ADD_VECTOR_WIDTH. SF_RN_ADD_VECTOR_WIDTH has a minimum value of 0 and a maximum value of 127. The total number of bits in the SF RN vector must not exceed 128 bits. The total can be calculated as $(\text{NUM_LOCAL_RNF} + \text{NUM_REMOTE_RNF}) / \text{SF_NUM_RNF_PER_CLUSTER} + \text{SF_RN_ADD_VECTOR_WIDTH}$.

The HN-F has the following fixed parameter:

- HN-F CHI interface data-VC (DAT) width is 256 bits.

4.2.2 Snoop connectivity and control

Each HN-F can send three types of snoop.

The available types of snoop request are:

Directed

To one RN-F

Multicast

To more than one RN-F but not all. If SF clustered mode is enabled, multicast snoops might be more common.

Broadcast

To all RN-Fs.

When SF clustering is enabled, HN-F may utilize CHI-E SnpQuery opcode. It can be sent to an RN-F that requests MakeReadUnique to HN-F. This helps HN-F determine the cache line presence RN-F at the time of processing the transaction.

4.2.3 Realm Management Extension support

The HN-F supports *Realm Management Extension* (RME) by treating the *Non-Secure* (NS) and *Non-Secure extension* (NSE) bits from a request as part of the address.

RME enables the HN-F to treat Secure, Non-Secure, Root, and Realm as four different areas of the memory space:

- The *Non-Secure* (NS) and *Non-Secure extension* (NSE) bits are stored in the SLC and SF tags.
- Snoops also propagate the NS and NSE bits as part of the message.
- Any request to the memory controller also propagates the NS and NSE bits.

4.2.4 HN-F SAM configuration by SN type

CMN-700 supports multiple SN types. You must program the HN-F SAM according to the types of SN that the HN-F targets.

You can configure CMN-700 to use the following SN types:

- SBSX
- CHI-C, CHI-D, CHI-E, or CHI-F SN-F
- CXL.mem

To configure SN target types in the HN-F SAM, program the `cmn_hns_sam_sn_properties` register. For the description of this register, see [5.2.4.94 cmn_hns_sam_sn_properties](#) on page 453.

The following table shows the bit values that you must program for each SN type.



The value of <x> describes the specific SN target.

Table 4-1: `cmn_hns_sam_sn_properties` SN type values

SN type	<x>_sn_is_chic	<x>_sn_is_chie	<x>_sn_is_chif	<x>_sn_pcmosep_conv_to_pcmo
CHI-C SN-F	0b1	0b0	0b0	0b1

SN type	<x>_sn_is_chic	<x>_sn_is_chie	<x>_sn_is_chif	<x>_sn_pcmosep_conv_to_pcmo
CHI-D SN-F	0b0	0b0	0b0	0b0
CHI-E SN-F	0b0	0b1	0b0	0b0
SBSX	0b0	0b1	0b0	0b0



Note

- If the AXI memory controller does not support PCMO on AW channel, then you must program HN-F as CHI-C SN mode.
- If the system uses CHI-E GrpIDExtn bits and the SN-F is CHI-B, CHI-C, or CHI-D, then you must set sn<x>_pcmo_conv_to_pcmo for HN-F to 0b1.
- If all SN types in the system do not support MTE, you must set cmn_hns_cfg.hnf_mte_mode_dis to 1.

4.2.5 Memory address decode error handling

HN-F SAM can detect Memory address decode error when an address is not programmed to target any SN.

For more information about the HN-F SAM, see [3.4.3 HN-F SAM](#) on page 54.

When HN-F encounters memory address decode error,

- The error is logged into HN-F RAS registers. See [3.5.2 HN-F error handling](#) on page 88
- HN-F completes CHI transactions internally without sending any SN requests
- HN-F issues broadcast snoops to clean up upstream copies if required
- Read Transactions:
 - If posion_on_mem_addr_dec_err_en is set, HN-F returns all 0s in data with poisons set; otherwise, HN-F returns all 1s in data without poisons.
- Write transactions:
 - Write data is dropped without allocating to SLC
 - Any SLC victims are dropped without writing back to SN



Note

HN-F ignores Memory address decode error if the address belongs to OCM region.

4.2.6 Hardware-based cache flush engine

The HN-F supports a hardware-based cache flush engine mechanism to flush the SF and SLC. The flush engine ensures that all cache lines in the lower and upper range are flushed from the CMN-700 SF and SLC.

Various *Address-Based Flush* (ABF) configuration registers per HN-F instance support the cache flush engine:

cmn_hns_abf_lo_addr

ABF lower range address.

cmn_hns_abf_hi_addr

ABF upper range address.

cmn_hns_abf_pr

ABF Policy Register. Triggers flush start, indicates flush operation type.

cmn_hns_abf_sr

ABF Status Register. Indicates flush completion and other status information.

The flush engine ensures that all cache lines in the lower and upper range are flushed from the CMN-700 SF and SLC. When all cache lines within this range are flushed, a bit in the `cmn_hns_abf_sr` register is set indicating that the flush engine has completed. If enabled, an interrupt, INTREQPPU, is then sent.



The interrupt indication and complete bit in the `cmn_hns_abf_sr` registers is set regardless of normal completion or abort condition. To determine if a flush request completed normally or aborted, check the error bits in the `cmn_hns_abf_sr` register.

To complete the flush sequence, the HN-F performs the following steps:

1. Flush CMN-700 SFs. This operation flushes the lines in the lower-level caches. Lower-level write-backs go to memory and are not allocated to the CMN-700 SLC.
2. Flush the CMN-700 SLC.
3. On completion of the flush:
 - a. The HN-F sets the status bit in the `cmn_hns_abf_sr` register when the flush is complete for that HN-F. If there are error conditions, they are also set in the `cmn_hns_abf_sr` register. This register is cleared when the next ABF request starts.
 - b. The HN-F sends a completion message to the global Power/Clock/Reset unit, and an optional INTREQPPU is asserted when all HN-F instances have completed the flush.

ABF requests are processed in parallel to other ongoing requests from RNs. If an ABF request and another ongoing request target the same address, then no ordering or coherency guarantee is provided. Power management transition requests have higher precedence than ABF requests. An ABF request is only supported when the Power management state is in FAM, HAM, or SFONLY mode and the retention state is IDLE or RETENTION, not transitional. While ABF is in progress,

any update to the *Power Policy Register* (PWPR) causes the ABF state machine to abort and the Power management request proceeds.

By default, the flush engine writes back any modified data to memory before invalidating the cache line from internal caches. Two more configuration modes are provided for user flexibility. Therefore, the flush engine has the following three modes of operation:

CleanInvalid	Write back and invalidate. This mode is the default.
MakeInvalid	In this mode, modified data is not written back to memory.
CleanShare	In this mode, modified data is written back to memory but clean data remains in internal caches.

If *On Chip Memory* (OCM) is enabled and the address range overlaps with the ABF range, OCM behavior supersedes ABF. For SLC, this condition means that for CleanInvalid and CleanShare modes, no action is taken. For MakeInvalid, there is no difference in behavior, regardless of whether the address is in the OCM range or not. For SF flush, the behavior is the same between an OCM address match and not.

The following tables summarise SF and SLC caches for all three modes.

Table 4-2: 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 4-3: 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)



The following assumptions are made:

- To ensure that coherency and ordering is maintained, RNs should not access a cache line within the flush range while ABF is in progress.
- Address ranges and trigger must be programmed for each HN-F in the SCG.

- Do not change ABF-related configuration register bits when the `abf_enable` bit of the `cmn_hns_abf_pr` register is set, until the `abf_complete` bit of the `cmn_hns_abf_sr` register indicates that the flush is done.
 - HN-F must be in one of the three operational modes, FAM, HAM, or SFONLY. When flush starts, any update to the PWPR causes ABF to abort.
 - SF must be enabled for the flush engine to operate. If SF is disabled, the flush engine aborts and indicates an error status in the `cmn_hns_abf_sr` register.
 - When ABF completes, check the `cmn_hns_abf_sr` to ensure that ABF completed without any errors. If ABF aborted for any reason, then the `cmn_hns_abf_sr` indicates that the flush was aborted.
-

4.2.7 Software-configurable memory region locking

The HN-F supports variable-size memory regions that can be locked in the system-level cache with way reservation.

These variable-size memory regions ensure that locked lines are not evicted from the SLC, and any access to those lines is guaranteed to hit in the SLC. This guarantee applies to normal-memory cacheable and allocatable requests as described in CHI specifications. Non-allocating type requests might still be pushed to memory. The variable memory region is calculated as a factor of the total SLC size and number of ways that are locked.

For example, if the SLC is built with 16 ways, then way locking of 1, 2, 4, 8, or 12 yields 1/16, 2/16, 4/16, 8/16, or 12/16 of the SLC size respectively.

Software uses the following mechanism to program the HN-F configuration registers to enable region locking:

- The `hnf_slc_lock_ways` register specifies the total number of locked HN-F system level cache ways. This register can have a value of 1, 2, 4, 8, or 12.
- The following region base registers specify the base address of the region that is using locked ways:
 - `hnf_slc_lock_base0` register
 - `hnf_slc_lock_base1` register
 - `hnf_slc_lock_base2` register
 - `hnf_slc_lock_base3` register
- A combination of the total SLC size, `hnf_slc_lock_ways` register, and the `hnf_slc_lock_base0` register to `hnf_slc_lock_base3` register defines the following:
 - The total amount of cache that is locked, calculated as follows:

Figure 4-1: Total cache locked equation

$$\frac{\text{Total SLC size} \times \text{Number of locked ways}}{16}$$

- Ways are locked beginning with way 0 and then in ascending order
- The number of valid regions and exactly which regions, and therefore which of the hnf_slc_lock_base0 to hnf_slc_lock_base3 registers, are valid and included in the HN-F way allocation
- The exact location, size, and alignment requirement of each region
- The region alignment is identical to the region size, for example:
 - A 0.5MB region is aligned to any 0.5MB boundary
 - A 4MB region is aligned to any 4MB boundary
- The size and alignment requirement is enforced in hardware, to prevent any errors in software
- Regions can be disjointed or contiguous, to create a larger single region
- All valid regions use all locked ways. There is no application-level way segregation.
- The HN-F must be in the FAM power state. Memory Region Locking is not supported in other CMN-700 power states.



The locked regions do not comprehend Secure, Non-Secure, Root, versus Realm memory regions. If aliasing is performed between any of the above regions, overlocking can occur.



SLC way locking is supported up to 128 HN-F in a single SCG.

The following tables specify various combinations of region size and the number of locked ways that software must program using the hnf_slc_lock_ways register and the hnf_slc_lock_base0 register to hnf_slc_lock_base3 register, based on an example configuration with an SCG total SLC size of 8MB.

Table 4-4: 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

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	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 4-5: 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]

4.2.8 Software-configurable On-Chip Memory

The CMN-700 HN-F supports software configurable *On-Chip Memory* (OCM). This enables the creation of systems without physical DDR memory. It also allows a system to use SLC as scratchpad memory.

In OCM mode, the HN-F does not send requests to the SN-F. To enable OCM, the following requirements must be met:

- The HN-F must be in the FAM power state. Other CMN-700 power states are not supported in OCM mode.
- All OCM ways must be the same across all HN-Fs in a system cache group.

In OCM mode, the following CMOs terminate in the SLC:

- CleanInvalid and CleanShared CMOs terminate in the SLC without invalidation or performing a WriteBack to the SN-F.
- MakeInvalid invalidates the cache line in SLC, and can be used to invalidate the OCM region.

OCM mode can be enabled by programming the hnf_ocm_en bit in the cmn_hns_cfg_ctl register. If the hnf_ocm_allways_en bit is set to 1, then all transactions targeting the HN-Fs have OCM behavior. The OCM region must be contiguous and aligned to the total SLC size of the configuration when the hnf_ocm_allways_en is set to 1. If the hnf_ocm_allways_en bit is 0, region locking registers define the OCM regions. For more information about these region locking registers, see [4.2.7 Software-configurable memory region locking](#) on page 128.



Note

Region locking registers do not explicitly control Secure, Non-Secure, Root, and Realm memory regions. Therefore, combined Secure, Non-Secure, Root, and Realm memory regions must not exceed the total SLC size that is locked for OCM.

CMN-700 HN-F supports dynamic OCM mode entry and exit.

Restrictions:

- Cannot be used as a full OCM mode (hnf_ocm_allways_en must be 1'b0).
- Can only be used in 1, 2, 4, 8, or 12 way locking.
- If OCM all ways is enabled, then all incoming addresses map to OCM, even if the address is outside the SLC size range.
 - The OCM region copy would not function because the target memory range never goes to MC.

Instructions for entering OCM mode:

1. Stop all RN traffic to HN-Fs.
2. Transition HN-F from FAM to SFONLY operating mode as [3.3.4 HN-F power domains](#) on page 22 describes.
3. Enable OCM mode using config registers.
4. Transition HN-F from SFONLY to FAM operating mode.
5. Enable traffic to HN-F including to OCM region(s).

Instructions for exiting OCM mode:

1. Stop all traffic from RNs to OCM region(s).
2. Using system software, copy the SLC contents of the OCM address region to memory using new, separate address space.
3. Initiate ABF in Makeinvalid mode to clear the OCM region(s) while OCM mode is still enabled..
4. When ABF is complete, disable OCM.

4.2.9 Source-based SLC cache partitioning

HN-F supports a feature to lock SLC ways for requesting nodes RN-F, RN-I, and RN-D.



The HN-F must be in the FAM power state.

This feature is an extension of the address-based way locking but the locked ways are based on the requestors instead of the programmed address. CMN-700 supports programming the number of ways that can be locked, RN devices for which these ways are locked, and allocation policies in each HN-F. With this feature enabled, the locked SLC ways are only available to the programmed RNs for any new cache line allocations.

You can enable the source-based way locking featured by programming the cmn_hns_rn_region_lock.rn_region_lock_en bit to 0b1 in each HN-F instance. You must also explicitly enable the requesting nodes in the cmn_hns_rn*region_vec registers for which these ways are to be locked. The requesting nodes are individually identified using the logical IDs or

cluster IDs. See [4.2.14 Configuring clustered mode for SF tracking](#) on page 136. Each requesting node type RN-F, RN-I, and RN-D has different registers. They are uniquely identified in the CMN-700 system using logical IDs. The number of ways that are locked are programmed in the `cmn_hns_slc_lock_ways.ways` field.

In configuration with up to 32 RN-I and RN-D, each bit represents the RN-I or RN-D logical ID. If the configuration contains greater than 32 RN-I and RN-D but less than 64, then LSB bit of the logical ID is dropped when indexing the region lock registers. Similarly if the configuration contains more than 64 RN-I and RN-D, then the least significant 2 bits of logical ID are dropped when indexing into the region lock configuration registers.

The region locking feature has the following allocation modes:

Allocating new cache lines for matching RNs only in the locked ways

The matching RNs are restricted to allocate to the locked partition only. Enable this mode by setting the `cmn_hns_rn_region_lock.rn_pick_locked_ways_only` bit to 0b1.

Allocating new cache lines for matching RNs to one of the locked or unlocked ways

This mode is the default behavior. In this mode, the locked ways are restricted only to the matching RNs but the unlocked ways are accessible by all the RNs.

Source-based SLC cache partitioning is supported only when *Enhanced LRU* (eLRU) mode is used.

Source-based cache partitioning is not supported:

- In other CMN-700 power states
- When the `CHI_MPAM_ENABLE` parameter is 'True'.

4.2.10 Way-based SLC cache partitioning

You can partition each SLC cache instance into different regions. This partitioning enables each requesting node (RN-F, RN-I, RN-D) to allocate in one or more regions, each consisting of four consecutive ways.

Each region group has configuration registers that indicate which of the logical RN-F, RN-I, or RN-D requesters can allocate to the corresponding group of SLC ways. By default, all RNs can allocate all 16 ways, as the following tables show.

Table 4-6: Logical RN-F ID requesting node

Register	Address	Ways reserved	Default	Logical RN-F ID								
	offset			63	62	61	60	...	3	2	1	0
cmn_hns_slcway_partition0_rnf_vec	0xC48	[3:0]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition1_rnf_vec	0xC50	[7:4]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition2_rnf_vec	0xC58	[11:8]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition3_rnf_vec	0xC60	[15:12]	{64'{1'b1}}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1

Table 4-7: Logical RN-I ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-I ID								
				31	30	29	28	...	3	2	1	0
cmn_hns_slcway_partition0_rni_vec	0xC68	[3:0]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition1_rni_vec	0xC70	[7:4]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition2_rni_vec	0xC78	[11:8]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1
cmn_hns_slcway_partition3_rni_vec	0xC80	[15:12]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1

Table 4-8: Logical RN-D ID requesting node

Register	Address offset	Ways reserved	Default	Logical RN-D ID									
				31	30	29	28	...	3	2	1	0	
cmn_hns_slcway_partition0_rnd_vec	0xC88	[3:0]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1	
cmn_hns_slcway_partition1_rnd_vec	0xC90	[7:4]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1	
cmn_hns_slcway_partition2_rnd_vec	0xC98	[11:8]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1	
cmn_hns_slcway_partition3_rnd_vec	0xCA0	[15:12]	{32'1'b1}	0/1	0/1	0/1	0/1	...	0/1	0/1	0/1	0/1	

The registers in these tables mask the ways that are available for an RN to allocate to at all times:

- 0b1** Indicates that the corresponding Logical RN ID or cluster ID can allocate in this region.
For more information about SF clustered mode and cluster IDs, see [4.2.14 Configuring clustered mode for SF tracking](#) on page 136.
- 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 4-1: Reserve ways 0-3 for RN-F {0-3}

- Write 64'h000000000000000F to cmn_hns_slcway_partition0_rnf_vec. This operation enables logical RN-F IDs 0, 1, 2, and 3 to allocate to ways 0-3. All other RN-F IDs (4-63) cannot allocate to these ways.
- Write 32'h0 to cmn_hns_slcway_partition0_rni_vec. This operation disables all RN-Is from allocating to ways 0-3.
- Write 32'h0 to cmn_hns_slcway_partition0_rnd_vec. This operation disables all RN-Ds from allocating to ways 0-3.

The following conditions apply to this example:



Note

- This feature cannot be used if region-based locking, source-based locking, or OCM is enabled.
- The region registers can be changed at runtime.
- When the way partitioning scheme is not being used, the preceding registers must be returned to the default values.

- Each RN must be configured to allocate to at least one partition. Setting the bit for a given RN to 0 in all partition registers defaults it to allocating in any partition.
- RN-I and RN-D each support a maximum logical ID of 96 using 32-bits in the partition mask register. In configurations with up to 32 RN-Is and RN-Ds, each bit represents the RN-I or RN-D logical ID. If the configuration contains greater than 32 RN-Is or RN-Ds but less than 64, then LSB bit of the logical ID is dropped when indexing the partition mask register. Similarly if the configuration contains more than 64 RN-Is and RN-Ds, then the least significant 2 bits of logical ID are dropped when indexing into the partition configuration register. In CML configurations, take care to assign LDID to remote RN-I and RN-Ds above the local RN-I and RN-D logical IDs. This requirement ensures that SLC partitioning is honored correctly across all RN-Is and RN-Ds.
- If SF clustered mode is enabled, HN-F uses the cluster ID instead of the full logical ID of the RNs. Therefore, all the RNs within a cluster can allocate to the locked ways.

Way-based SLC cache partitioning is supported only when *Enhanced LRU* (eLRU) mode is used.

Way-based SLC cache partitioning is not supported:

- In other CMN-700 power states
- When the `CHI_MPAM_ENABLE` parameter is 'True'.



The HN-F must be in the FAM power state.

4.2.11 RN-F tracking in the SF

The CMN-700 HN-F SF tracks cache lines that come from specific RN-Fs using an RN-F vector index. Your CMN-700 configuration and the values of some configurable parameters determine the size of this width.

The maximum width of the vector index is 128 bits.

Use the following parameters to calculate the total width of the SF vector:

$$SF_TOTAL_WIDTH = SF_MIN_WIDTH + SF_RN_ADD_VECTOR_WIDTH$$

The configuration calculates `SF_MIN_WIDTH`, using the following equation:

$$SF_MIN_WIDTH = \text{Ceil}((NUM_LOCAL_RNF + NUM_REMOTE_RNF) / SF_MAX_RN_PER_CLUSTER)$$

The `SF_MIN_WIDTH` calculation ensures that the SF vector index can track all the local and remote RN-Fs in your configuration. However, you can also configure the SF vector index at build time to contain extra bits by using the `SF_RN_ADD_VECTOR_WIDTH` parameter. This parameter lets you add extra bits to the SF vector, up to a combined maximum `SF_TOTAL_WIDTH` of 128 bits.

You can choose between two modes for tracking RN-Fs in the SF, or you can use a mixture of the modes. For more information about these modes, see [4.2.12 Non-clustered and clustered modes for SF RN-F tracking](#) on page 135.

4.2.12 Non-clustered and clustered modes for SF RN-F tracking

The CMN-700 SF uses non-clustered and clustered mode to track cache lines from RN-Fs. The mode that you use affects the number of RN-Fs that the SF can track.

Non-clustered mode

Each entry in the RN-F vector index is associated with a single RN-F. In this mode, CMN-700 is limited to a maximum of 128 RN-Fs. This limitation is because the SF can only track cache lines from a maximum of 128 RN-Fs. This limitation also applies to CML configurations.

Clustered mode

Each entry in the RN-F vector index is associated with a group of RN-Fs. You can group two, four, or eight RN-Fs into a cluster using this mode. This mode lets you increase the number of RN-Fs in a system up to a maximum of 512. The number of clusters is limited to 128 in this mode.

Both non-clustered and clustered mode use RN-F LDIDs to map RN-Fs to entries in the index. However, the way that the SF uses LDIDs to map RN-Fs to index entries differs between the two modes. In non-clustered mode, the whole LDID is used to map an RN-F to a single entry. In clustered mode, part of the LDID indicates which cluster group an RN-F belongs to and another part indicates the device within the group. In this case, the first part of the LDID is known as the cluster ID, and the second part is known as the device ID. Each bit in the SF vector represents the cluster ID of the RNs. For example, if there are four RN-Fs in a cluster, the SF is addressing all four RN-Fs. For any shared lines, HN-F snoops all four RN-Fs in the cluster.

For more information about the modes, including example configurations, see the following sections:

- [4.2.13 Configuring non-clustered RN-F tracking in HN-F SF](#) on page 135
- [4.2.14 Configuring clustered mode for SF tracking](#) on page 136
- [4.2.15 Identifying clusters and individual devices in clustered mode](#) on page 136

4.2.13 Configuring non-clustered RN-F tracking in HN-F SF

To enable non-clustered mode, set `SF_MAX_RNF_PER_CLUSTER` parameter to 1. The LDID of each RN-F in the system is assigned to a single index entry in the SF RN-F vector index.

For example, the following table shows an example of mapping between the RN-F vector index and the RN-F LDIDs in a configuration with 128 non-clustered RN-Fs.

Table 4-9: 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.

4.2.14 Configuring clustered mode for SF tracking

You can enable clustered mode for RN-F tracking in the SF by setting the `SF_MAX_RNF_PER_CLUSTER` parameter value to a valid value that is >1. This parameter specifies the maximum number of RN-Fs that are in a cluster group, up to a maximum of eight.

If clustered mode is enabled, there must be at least two cluster groups. You cannot cluster all RN-Fs into a single cluster group.

If there is a single RN-F in the cluster, then it must always use the lowest device ID in that cluster. Consider a four-way clustering, where the device ID fields are 0b00, 0b01, 0b10, and 0b11. In this configuration, you cannot use device ID values of 0b01, 0b10, or 0b11 unless 0b00 is also used.

You can combine local and remote RN-Fs into the same cluster. Each cluster can contain any number of RN-Fs, up to the value of the `SF_MAX_RNF_PER_CLUSTER` value. For example, in a system with `SF_MAX_RNF_PER_CLUSTER` = 4, cluster 0 could contain four RN-Fs and cluster 1 could contain only two RN-Fs.

The size of the SF RN-F vector index is partly configurable, as [4.2.11 RN-F tracking in the SF](#) on page 134 describes. You can use the `SF_RN_ADD_VECTOR_WIDTH` parameter to set up a mixed clustered and non-clustered tracking scheme. This type of scheme enables the SF to track some RN-Fs clustered in groups, and others individually.

When you choose to use clustered mode, also consider the following:

- SLC partitioning and SLC way locking using source ID use the cluster ID instead of the full LDID. Therefore, all RN-Fs in the cluster can use the partitioned ways for this source ID.
- Using clustered mode in the CMN-700 SF increases the likelihood of SF back in-invalidations if the number of sharers exceeds the maximum precise tracking ability of the SF vector. 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.

4.2.15 Identifying clusters and individual devices in clustered mode

For the SF to track and identify RN-Fs within cluster groups, the RN-F LDID bits are divided into separate components. These components are known as the cluster ID and the device ID.

The number of LDID bits that these components use depends on the following properties of your configuration:

- The number of RNs in the system that are addressable by the SF RN-F vector index. The width of the LDID scales up to a maximum size of 9 bits in a system with 512 RN-Fs.
- The value of the `SF_MAX_RNF_PER_CLUSTER` parameter.

Calculate the width of the device ID component of the LDID using the following equation:

$$\text{Device ID width} = \log_2(\text{SF_MAX_RNF_PER_CLUSTER})$$

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 4-10: Cluster ID and device ID LDID components for 512 RN-F configuration

MAX_RNF_PER_CLUSTER value	LDID ranges	Clustering components
4	LDID[8:2]	Cluster ID[6:0]
	LDID[1:0]	Device ID[1:0]
8	LDID[8:3]	Cluster ID[5:0]
	LDID[2:0]	Device ID[2:0]

You must consider the cluster ID and device ID when assigning LDIDs to local and remote RN-Fs. Accounting for these values helps you ensure that RN-Fs are grouped in the cluster as required.

Example cluster mode configurations

The following table shows an example configuration with 512 RN-Fs, a 128-bit RN-F vector index, and cluster groups of four RN-Fs.

Table 4-11: Example clustered mode configuration with four RN-Fs per cluster group

RN-F vector index value (cluster ID)	Device ID 0 (LDID)	Device ID 1 (LDID)	Device ID 2 (LDID)	Device ID 3 (LDID)
0	LDID_0	LDID_1	LDID_2	LDID_3
1	LDID_4	LDID_5	LDID_6	LDID_7
2	LDID_8	LDID_9	LDID_10	LDID_11
...
127	LDID_508	LDID_509	LDID_510	LDID_511

The following table shows an example configuration with 512 RN-Fs, a 64-bit RN-F vector index, and cluster groups of eight RN-Fs.

Table 4-12: Example clustered configuration with eight RN-Fs per cluster group

RN-F vector index value (cluster ID)	Device ID 0 (LDID)	Device ID 1 (LDID)	Device ID 2 (LDID)	Device ID 3 (LDID)	Device ID 4...6	Device ID 7 (LDID)
0	LDID_0	LDID_1	LDID_2	LDID_3	...	LDID_7
1	LDID_8	LDID_9	LDID_10	LDID_11	...	LDID_15
2	LDID_16	LDID_17	LDID_18	LDID_19	...	LDID_23
...
63	LDID_504	LDID_505	LDID_506	LDID_507	...	LDID_511

You can use SF clustering in the following ways:

Fully precise mode

Each cluster only contains one RN-F. This mode behaves similar to non-clustered mode.

Hybrid mode

Each cluster can contain zero to `SF_MAX_RNF_PER_CLUSTER` RN-Fs

Fully clustered mode

Each cluster contains maximum RN-Fs up to `SF_MAX_RNF_PER_CLUSTER` parameter.

Consider a CMN-700 system with 8 RN-Fs and 4-way SF clustering is enabled in HN- F (`SF_MAX_RNF_PER_CLUSTER` = 4). The SF vector width is calculated as:

Minimum SF width = $((LOCAL_RNF + REMOTE_RNF) / SF_MAX_RNF_PER_CLUSTER)$

The result of the calculation translates to SF tracking vector with 2 bits. A system builder can choose to have extra tracking bits in the SF using the `SF_RN_ADD_VECTOR_WIDTH` parameter. For example, if this parameter is set to 6, then the total SF tracking ability is extended to 8 clusters (`TOTAL_SF_VEC_WIDTH` = 8) with 4 RN-Fs each. Therefore, up to 32 RN-Fs can be tracked in the SF with LDIDs 0-31. In this configuration, the `LDID_WIDTH` = 5.

The following table shows the LDID mapping to cluster and device ID as per the SF tracking vector.

Table 4-13: LDID mapping to SF cluster

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	LDID2	LDID3
1	LDID4	LDID5	LDID6	LDID7
2	LDID8	LDID9	LDID10	LDID11
3	LDID12	LDID13	LDID14	LDID15
4	LDID16	LDID17	LDID18	LDID19
5	LDID20	LDID21	LDID22	LDID23
6	LDID24	LDID25	LDID26	LDID27
7	LDID28	LDID29	LDID30	LDID31

Each LDID entry in the preceding table contains RN-F attributes such as:

RN_Valid

Indicates a valid RN-F is mapped to this LDID.

CML Port Aggregation (CPA) enable

If the RN-F is a remote RN-F, this attribute indicates that *CML Port Aggregation (CPA)* is enabled.

CML Port Aggregation Group (CPAG) ID

If this RN-F is a remote RN-F, this attribute indicates the *CML Port Aggregation Group (CPAG)* ID.

Remote

To indicate this RN-F is a remote RN-F.

CHI protocol supported

The protocol version of this RN-F (CHI-B/C/D/E).

RN NodeID

Physical NodeID of this RN-F. For remote RN-Fs, this attribute can contain the CCG nodeID.

These fields are programmed in cmn_hns_rn_cluster[0-127]_physid_reg[0-3] register in HN-F. It is used for HN-F to send snoop requests.

Default LDID mapping to SF clustering table

Out of reset, MXP assigns sequential LDIDs to RN-Fs within each chip. If all RN-Fs are on the same chip, they are assigned LDID 0-7. You can program the attributes of each LDID as the following table shows. With the default LDIDs, only clusters 0 and 1 are used and clusters 2-7 are unused.

Table 4-14: Default LDID mapping to SF clusters

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	LDID2	LDID3
1	LDID4	LDID5	LDID6	LDID7
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7	-	-	-	-

In this mode, SF is tracking the RN-Fs at cluster level for shared cache lines. Therefore, tracking is imprecise and HN-F can snoop all the RN-Fs in that cluster when required.

Fully precise mode

Each cluster can be programmed to have one RN-F so that it can be precisely tracked in the SF. In this mode, default LDIDs in MXP must be programmed to match the cluster to LDID mapping the following table shows.

Table 4-15: Precise mode LDID mapping to clusters

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	-	-	-
1	LDID4	-	-	-
2	LDID8	-	-	-
3	LDID12	-	-	-
4	LDID16	-	-	-
5	LDID20	-	-	-
6	LDID24	-	-	-
7	LDID28	-	-	-

In this mode, every RN-F is precisely tracked in the SF vector.

Hybrid mode

In hybrid mode, the number of RN-Fs in each cluster can be different. In other words, the SF tracks RN-Fs with a mix of precise and imprecise tracking. You must program the default LDIDs in MXP to match the following table.

Table 4-16: Hybrid LDID mapping to SF clusters

SF tracking vector ID (cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	-	-
1	LDID4	-	LDID6	-
2	LDID8	-	-	-
3	LDID12	-	-	-
4	LDID16	-	-	-
5	LDID20	-	-	-
6	-	-	-	-
7	-	-	-	-

In this mode, HN-F tracks cluster 0 and 1 in imprecise mode for shared cache lines. Clusters 2-5 are tracked in precise mode.

Fully clustered mode

In fully clustered mode, every cluster contains as many valid RN-Fs as when configured using the `SF_MAX_RNF_PER_CLUSTER` parameter. For example, in a system with 32 RN-Fs and `SF_MAX_RNF_PER_CLUSTER = 4`, the SF width = 8. Therefore, the LDID to cluster mapping is fully populated.

Table 4-17: Fully clustered mode mapping of LDID to SF clusters

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
0	LDID0	LDID1	LDID2	LDID3

SF tracking vector ID (Cluster ID)	Device ID 0	Device ID 1	Device ID 2	Device ID 3
1	LDID4	LDID5	LDID6	LDID7
2	LDID8	LDID9	LDID10	LDID11
3	LDID12	LDID13	LDID14	LDID15
4	LDID16	LDID17	LDID18	LDID19
5	LDID20	LDID21	LDID22	LDID23
6	LDID24	LDID25	LDID26	LDID27
7	LDID28	LDID29	LDID30	LDID31

In this mode, shared cache lines are tracked at cluster level. HN-F can snoop all the RN-Fs in the cluster when required to resolve coherency.

For the 8-cluster examples, the following registers must be programmed for each LDID and they must be consistent within each chip.

HN-F registers:

For an example of 8 clusters with 4 RN-Fs each, you must program the following HN-F registers:

- `cmn_hns_rn_cluster0_physid_reg[0-1]`
- `cmn_hns_rn_cluster1_physid_reg[0-1]`
- `cmn_hns_rn_cluster2_physid_reg[0-1]`
- `cmn_hns_rn_cluster3_physid_reg[0-1]`
- `cmn_hns_rn_cluster4_physid_reg[0-1]`
- `cmn_hns_rn_cluster5_physid_reg[0-1]`
- `cmn_hns_rn_cluster6_physid_reg[0-1]`
- `cmn_hns_rn_cluster7_physid_reg[0-1]`

These registers contain attributes for each RN-F such as its NODEID, SrcType, CPA info, remote or local, and a valid bit.

MXP registers:

Default LDID assignment of RN-Fs attached to each MXP device port p[0-3] (directly or behind CAL) are captured in the register:

- `por_mxp_device_port_connect_ldid_info_p[0-3]`: Read-only register

In hybrid mode and precise modes, if you change the default LDID, you must program the MXP LDID override registers appropriately:

- `por_mxp_p[0-3]_ldid_override`
 - Program these registers to override the LDID assignment of RN-F or RN-Fs attached (directly or behind CAL) to each MXP device port p[0-3].
 - Reset value of these registers is the default LDID assignment.

- These registers are present only if clustering is enabled and RN-F or RN-Fs are connected to this device port.

RA register:

Assign *Requesting Agent IDs* (RAIDs) for local RNs.

For each local RN, identified by their LDID, program RAIDs and set the corresponding valid bit in the following registers:

- por_ccg_ra_rnf_ldid_to_exp_raid_reg [0-127]
- por_ccg_ra_rni_ldid_to_exp_raid_reg [0-9]
- por_ccg_ra_rnd_ldid_to_exp_raid_reg [0-9]

This programming sets up the LDID to RAID LUT.

LDID override. Consider a situation where the LDID of a given RN-F is overridden by programming the corresponding register in MXP. In this case, the following registers at RA must be programmed with the overridden value for that particular LDID:

- por_ccg_ra_rnf_ldid_to_ovrd_ldid_reg [0-127]

HA registers:

Program unique LDIDs for each remote caching agent in the following register:

- por_ccg_ha_rnf_exp_raid_to_ldid_reg [0-255]

Set the ldid<X>_rnf bit, which marks the remote agent as a caching agent, and set the respective valid bit.

The LDID values for remote RN-Fs must be greater than those values that are used by the local RN-F nodes, unless RN-Fs LDIDs are overridden.

HN-F also supports precise tracking up to n-sharers when clustering is enabled. This mode is only available when $TOTAL_SF_VEC_WIDTH \geq 2 * (LDID_WIDTH + 1)$. That is, the SF RN-F tracking vector must have enough bits to track at least 2 LDIDs and a valid bit for each LDID. In the preceding example with 8 clusters and 4-way clustering, LDID_WIDTH is 5 bits wide. Therefore, precise mode is not available in this configuration.

If $SF_RN_ADD_VECTOR_WIDTH = 4$ in the preceding example, then precise mode is enabled, and it can track 2 sharers without having to snoop all RN-Fs within a cluster. This mode is enabled by default when a configuration has enough bits to track at least 2 RN-Fs. Disable this mode by programming the cmn_hns_cfg_ctl.sf_rnf_vec_precise_mode_en config bit to 1'b0. In the preceding example with 8 clusters and $SF_RN_ADD_VECTOR_WIDTH = 4$, the RN-F tracking vector in SF is 12 bits wide. In the precise tracking mode, the RN-F vector represents the decoded RN-F LDIDs as the following table shows.

Table 4-18: SF vector tracking RN-Fs in Precise mode

11	10	9	8	7	6	5	4	3	2	1	0
Second RN-F LDID[4:0]					Valid	First RN-F LDID[4:0]					Valid

When a cacheline is accessed the first time, the LDID of the requesting RN-F is tracked in the RN-F tracking vector with valid set to 1. Then, if a second RN-F accesses this same cache line, it occupies the second entry in the preceding table. Even if these 2 RN-Fs belong to separate clusters with 4 RN-Fs each, HN-F tracks them precisely. For example, consider a situation where the first RN-F is from cluster 2 and the second RN-F is from cluster 7. In this case, the preceding decode LDID vector fully tracks the true LDID of the requestors. In this mode, any evict operation from these RN-Fs is also precise and so there is no SF pollution.

Then consider a situation in which a third RN-F from cluster 9 accesses the same cache line. In this case, the SF switches from precise to imprecise mode because it does not have enough bits to track the third RN-F in precise mode. The RNFVEC is updated as the following table shows.

Table 4-19: Vector tracking RN-Fs in imprecise mode

11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	0	0	0	0	1	0	0

The imprecise mode is reset when HN-F receives either:

- A request that requires the cache line to migrate to RN-F, such as ReadUnique/MakeReadUnique
- Writes that require SF entry to be cleared

4.3 Error reporting and software-configured error injection

HN-F detects and reports several types of errors to the error block.

HN-F supports the following types of errors:

Correctable errors

For example, single-bit ECC error detection and correction in the SLC Tag RAM, SF Tag RAM, and SLC Data RAM

Deferred errors

For example, double-bit ECC error detection in SLC Data RAM

Uncorrectable errors

For example, double-bit ECC errors in the SLC or SF Tag RAMs

If the DATACHECK_EN parameter is enabled, HN-F can also support Data parity error detection in the SLC Data RAM. These errors are logged as Deferred Errors.

For logging and reporting all error types, HN-F follows the procedures described in [3.5 Reliability, Availability, and Serviceability](#) on page 85.

For information regarding the error source, see the ERRSRC field of [5.2.4.28 cmn_hns_errmisc0](#) on page 379 and [5.2.4.29 cmn_hns_errmisc1](#) on page 380.

4.3.1 Software-configurable error injection

The HN-F supports software-configurable error injection and reporting. This feature enables testing of the software error handler routine for SLC double-bit ECC data errors.

The HN-F configuration register for a particular logical thread enables configurable error injection and reporting.

Any Cacheable read for which the HN-F provides the data is defined as a system cache hit. If error injection and reporting are enabled, any system cache hit drives the Completer 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.



This mechanism is designed to mimic SLC data ECC errors for SLC hits. If enabled, this mechanism only causes an error to be logged and optionally an interrupt to be generated. SLC misses do not drive any Completer errors or error interrupts. Error injection on SLC hits does not alter the Resp*, Poison, or Data fields in the DAT flit that is returned to the RN.

For more information about configuring error injection, see [5.2.4.53 cmn_hns_err_inj](#) on page 407.

4.3.2 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 [5.2.4.54 cmn_hns_byte_par_err_inj](#) on page 408. This register specifies the byte lane from 0-31 in which a parity error is introduced. The memory data is uncorrupted with such injection, but the Data Check field of the DAT flit that is returned to an RN is altered.

4.4 Transaction handling in SLC memory system

The CMN-700 SLC memory system handles various types of CHI operations and transaction fields. The structure of the overall system and how each component is configured affects how these transactions are handled.

4.4.1 Cache maintenance operations

CMN-700 uses several CHI *Cache Maintenance Operations* (CMOs).

The following operations are supported:

- CleanInvalid

- CleanShared
- MakeInvalid
- CleanSharedPersist
- CleanSharedPersistSep

These operations always look up the SLC and the SF, and take the following actions:

- Clean and invalidate the line if present in the SLC.
- If the CMO is address hits in the SF, the HN-F sends a snoop to the RN-F post SF lookup if necessary.
- If the cache line is modified in the SLC or in the cache of the RN-Fs, the HN-F initiates a memory controller WriteBack if necessary.



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.

4.4.2 Cacheable and Non-cacheable exclusives

The HN-F supports PoC monitor functionality for Cacheable and Snooperable exclusive operations from the RN-Fs.

The Cacheable and Snooperable exclusive transactions are:

- ReadShared
- ReadClean
- CleanUnique
- ReadPreferUnique
- MakeReadUnique
- ReadNotSharedDirty

The HN-F also supports system monitor functionality for Non-cacheable exclusive support. For more information about exclusives, see the *AMBA® 5 CHI Architecture Specification*.



Each HN-F in CMN-700 can support tracking of up to 512 logical processors for non-Cacheable exclusive operations:

- In configurations with up to 64 RN-Fs, HN-F supports 64 exclusive monitors.
- In configurations with 64-144 RN-Fs, HN-F supports 144 exclusive monitors.

- In configurations above 144 RN-Fs, the total number of exclusive monitors is equivalent to the total number of local and remote RN-Fs, RN-Is, and RN-Ds, up to a maximum of 512 exclusive monitors.

For Non-cacheable exclusive, the system programmer must ensure that there are no more logical processors capable of concurrently sending exclusive operations than the number of exclusive monitors.

Cacheable exclusive is implemented based on Snoop Filter and not constrained by the number of exclusive monitors.

4.4.3 DataSource handling

CMN-700 populates the DataSource field of a CompData response to specify the source of the data.

DataSource information can be used:

- To determine the usefulness of a PrefetchTarget (Memory controller prefetch) transaction
- To profile and debug software to evaluate and optimize data sharing patterns

Table 4-20: DataSource encodings

Source of data	Message	Encoding
HN-I	Default (non-memory source)	0b0000
RN-F	Peer processor cache within local cluster	0b0001
RN-F	Local cluster cache	0b0010
HN-F	<i>System Level Cache (SLC)</i>	0b0011
RN-F	Peer cluster cache	0b0100
Remote chip	Remote chip caches	0b0101
SN-F or SBSX	PrefetchTarget was useful	0b0110
SN-F or SBSX	PrefetchTarget was not useful	0b0111
RN-F	Local cluster cache, unused prefetch	0b1010
HN-F	System Level Cache (SLC), unused prefetch	0b1011

CMN-700 drives the DataSource value only when the source of the data is either HN-F or HN-I. For other data sources, CMN-700 acts as a conduit.

The encoding that is used by CMN-700 to indicate a data source is the same as the suggested value in the *AMBA® 5 CHI Architecture Specification*. Any deviation from the specified encodings might result in unexpected behavior.

4.4.4 CMO and PCMO propagation from HN-F to SN-F or SBSX

CMN-700 supports propagation of CMO and PCMO requests for a given cache line to the memory controller. The completion point of these requests is programmable.

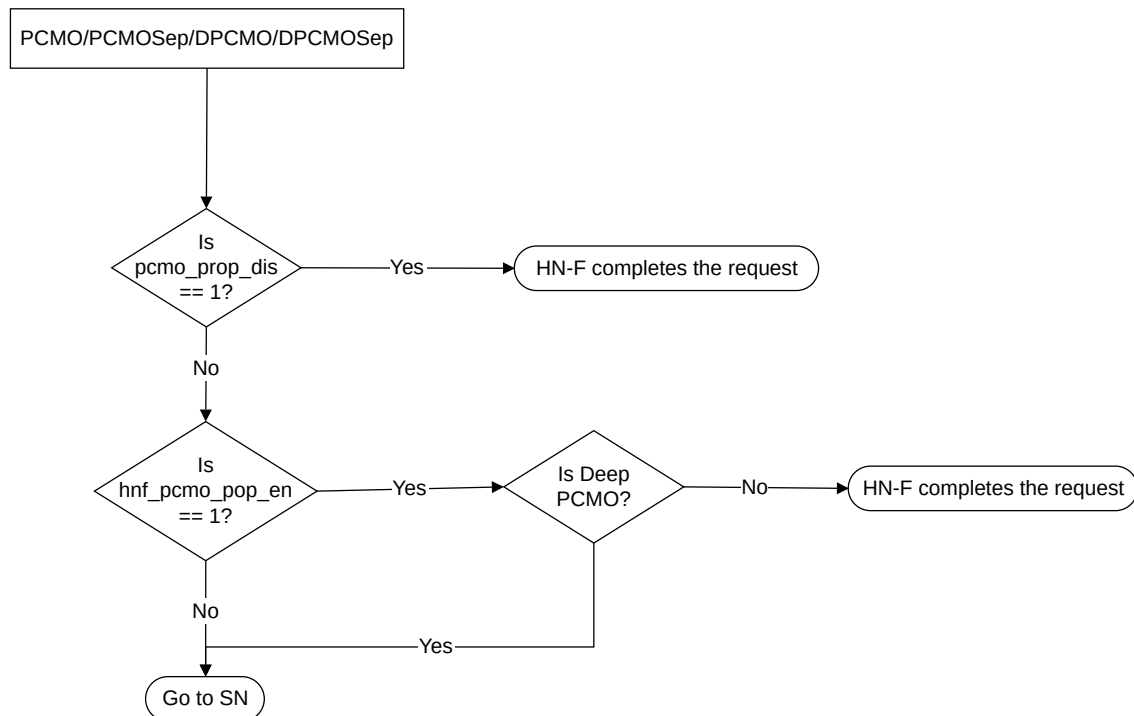
This feature ensures that the cache line has been written to the memory controller and that any copies in the CMN-700 system have been removed. Conditional CMO and PCMO propagation to the memory controller also supports external DRAM caches. You can enable or disable this feature in the `cmn_hns_sam_sn_properties` register bits for each HN-F corresponding to each SN-F. For SBSX with an AXI4 Completer memory device, you must disable CMO and PCMO propagation in HN-F.

The following HN-F SAM configuration register bits decide the CleanSharedPersist (PCMO) and CleanSharedPersistSep (PCMOsep) completion point:

- `cmn_hns_sam_sn_properties.pcmo_prop_dis`
- `cmn_hns_cfg_ctl.hnf_pcmo_pop_en`

You can program this behavior per SN. Deep is an attribute in the CHI-D request flit and applies to PCMO type requests (DPCMOs). HN-Fs use the Deep attribute and the HN-F SAM configuration bits to decide the PCMO request completion point. The following diagram shows the PCMO and PCMOsep request completion point decision process.

Figure 4-2: PCMO and PCMOsep completion point flow diagram



If an HN-F is programmed to propagate PCMO requests to an SN-F, and the SN-F does not support PCMOsep requests, you must set the `cmn_hns_sam_sn_properties.X_sn_pcmosep_conv_to_pcmo` bit to 0b1. This setting enables the HN-F to complete the request by converting PCMOsep type requests to PCMO. The HN-F generates a Persist response to the requestor on receiving a completion of PCMO response from SN-F.



CHI-C and CHI-D SN-Fs do not support the GroupIDExt field in the REQ flit. Therefore, if both of the following criteria are true for your CMN-700 system, you must program HN-Fs to terminate persistent CMOs:

- CMN-700 contains any CHI-C or CHI-D SN-Fs.
- RNs in the system can issue persistent CMOs.



If the `sbsx_cmo_on_aw` parameter = 0, you must program the `cmn_hns_sam_sn_properties.X_sn_pcmosep_conv_to_pcmo` configuration bit. Setting this bit prevents HN-F from sending PCMO opcodes that map to two-part PCMOs on AW to SBSX.

You can also program HN-Fs to propagate CleanShared, CleanInvalid, and MakeInvalid CMOs to an SN-F. The following table shows how to program this behavior using the `cmn_hns_sam_sn_properties` register.

Table 4-21: CMO propagation programming in HN-F SAM

HN-F SAM attribute (<code>cmo_prop_en</code>)	CMO completion point
0b0	HN-F
0b1	SN

See the following register descriptions for more information:

- [5.2.4.94 `cmn_hns_sam_sn_properties`](#) on page 453
- [5.2.4.7 `cmn_hns_cfg_ctl`](#) on page 347

4.4.5 Memory System Performance Resource Partitioning and Monitoring

CMN-700 supports *Memory System Performance Resource Partitioning and Monitoring* (MPAM). MPAM features enable software to optimize the use of memory resources and to monitor how those resources are used.

If MPAM is enabled, then an extra MPAM field is added to the REQ and SNP channels. This field must be stored and propagated to the downstream target.

For more information about MPAM, see the following documents:

- *AMBA® 5 CHI Architecture Specification*

- *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM), for Armv8-A*

4.4.5.1 MPAM propagation

When MPAM is enabled, CMN-700 propagates the MPAM fields throughout the network. Various nodes are designed to propagate the MPAM field when processing requests.

The following node types propagate the MPAM field:

- MXP
- RN-I
- HN-F and HN-I
- CML RA and CML HA
- SBSX



Note

The `AXMPAM_EN` parameter controls if MPAM bits are stored and propagated on RN-I, HN-I, and SBSX bridges.

If MPAM support is required in CMN-700, set the `CHI_MPAM_ENABLE` parameter.

When an HN-F node receives a request, it stores the details from the MPAM field. If the request misses in the SLC, then the HN-F must drive the stored MPAM values onto the outgoing request to the memory controller. The MPAM field contents are stored in the `SLC_TAG` array and propagated into requests that are sent to memory.

When MPAM is enabled, an extra register field, `*_mpam_override_en`, is added to the MXP. You can program this field to override the MPAM value of the RN-F request with the set value in the register. See .

When MPAM is enabled, the RN-I has the following extra components:

- Two extra signals on the AXI or ACE-Lite Completer port, `ARMPAM[10:0]` and `AWMPAM[10:0]`
- An extra register field, `*_mpam_override_en`, which you can program to override the request MPAM value with the set value in the register



Note

If `AXMPAM_EN = 0`, then the MPAM override is active by default.

The CML RA node type has the following MPAM modes:

SMP mode

The CML RA passes the MPAM field on the `USER` field of the request. When snooped, the CML RA receives the MPAM field and passes it through the `CHI SNP MPAM` field.

Non-SMP mode

The CML RA drops the MPAM field that is received on the CHI request. The CML RA also does not receive MPAM field on CML snoops in this mode.

CXSA mode

The CML RA passes the MPAM field on the USER field of the request, even though CXSA is in non-SMP mode. You can use a configuration bit to enable passing of MPAM attributes when in CXSA mode.

The CML HA node type has the following MPAM modes:

SMP mode

The CML HA receives MPAM fields through the USER field of the request and passes them through the CHI MPAM field. On incoming CHI transactions, the CML HA passes CHI MPAM values through on the USER field.

Non-SMP mode

The CML HA drops the MPAM values that it receives on CHI snoop. The CML HA does not receive the MPAM attributes on CML requests.

If `AXMPAM_EN` = 1, the HN-I and SBSX node types propagate CHI.RXREQ.MPAM onto the AXI pins. If `AXMPAM_EN` = 0, the HN-I and SBSX node types drive 0s onto AXI pins.

In CMN-700, MPAM support is applicable to the SLC.

4.4.5.2 MPAM configuration

CMN-700 provides several configuration parameters to configure MPAM features that are used by the interconnect. The MPAM feature ID register, `cmn_hns_mpam_idr`, provides information about the MPAM features that are supported in the design.

The number of partitions and the number of performance monitoring groups that are supported is configurable at build time. The default number of partitions is 64 for Non-Secure partitions and 16 for Secure partitions. The default number of performance monitoring groups is two.

MPAM Non-Secure and Realm CSUMON configurations are based on a combined total of CSUMONs (`HNS_MPAM_NS_NUM_CSUMON` + `HNS_MPAM_RL_NUM_CSUMON`)

3'b000	1 CSUMON allocated to Realm; Remaining CSUMONs allocated to Non-Secure
3'b001	1 CSUMON allocated to Non-Secure; Remaining CSUMONs allocated to Realm
3'b010	1/2 CSUMONs allocated to Non-Secure; 1/2 to Realm. Remaining 1 CSUMON allocated to Non-Secure
3'b011	3/4 CSUMONs allocated to Non-Secure; 1/4 to Realm. Remaining 1 CSUMON allocated to Non-Secure
3'b100	1/4 CSUMONs allocated to Non-Secure; 3/4 to Realm. Remaining 1 CSUMON allocated to Non-Secure

MPAM Non-Secure and Realm PARTID configurations are based on a combined total of PARTIDs ($\text{HNS_MPAM_NS_PARTID_MAX} + \text{HNS_MPAM_RL_PARTID_MAX}$)

3'b000	1 PARTID allocated to Realm; Remaining PARTIDs allocated to Non-Secure
3'b001	1 PARTID allocated to Non-Secure; Remaining PARTIDs allocated to Realm
3'b010	1/2 PARTIDs allocated to Non-Secure; 1/2 to Realm. Remaining 1 PARTID allocated to Non-Secure
3'b011	3/4 PARTIDs allocated to Non-Secure; 1/4 to Realm. Remaining 1 PARTID allocated to Non-Secure
3'b100	1/4 PARTIDs allocated to Non-Secure; 3/4 to Realm. Remaining 1 PARTID allocated to Non-Secure

For more information about supported MPAM features, see [5.2.3.3 cmn_hns_ns_mpam_idr](#) on page 265.

For more information about the software-programmable MPAM override mechanism, see [4.4.5.6 Software-programmable MPAM override](#) on page 153.

4.4.5.3 Cache portion and capacity partitioning

MPAM support in the interconnect is for the CMN-700 SLC. MPAM is supported only with an *enhanced Least Recently Used* (eLRU) cache replacement policy. When MPAM is enabled, eLRU mode is enabled automatically when either cache portion or cache capacity partitioning are enabled.

The SLC supports cache portion partitioning that is based on the following masks:

MPAMCFG_CPB

The CPBM value for a request PARTID determines which cache portions a request can allocate.

MPAMCFG_CMAX

The CMAX value for a request PARTID determines the percentage of the SLC that a request can use.

Cache portion and capacity partitioning have the following features:

- The number of portions is the same as the number of ways in SLC.
- The `hnf_mpam_ccap_idr_cmax_wd` field of the `cmn_hns_mpam_ccap_idr` register is set to 7. This setting provides granularity of 0.78% ($1 / 2^7$) SLC for cache capacity partitioning.
- In HAM mode, portions 15:8 are aliased to portions 7:0. Cache capacity is adjusted for half the cache.
- CMN-700 supports address-based locking, including OCM, with MPAM. Locked ways are not available for MPAM-based partitioning. Cache capacity is adjusted to account for locked ways.



CMN-700 does not support source-based or way-based SLC partitioning with MPAM.

If using way locking with MPAM, you must program the lock registers first. Then, to determine the number of available portions, read the `hnf_mpam_cpor_idr_cpbm_wd` field of the `cmn_hns_mpam_cpor_idr` register. Locked ways also reduce cache capacity.

HN-F MPAM counter values are not accurate when exiting retention state and can result in underflow conditions.

4.4.5.4 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 `cmn_hns_X_msmon_cfg_csuflt` and `cmn_hns_X_msmon_cfg_csuctl` registers determine filter and control for each monitor.



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_csuctl_capt_evtnt` field of the `cmn_hns_X_msmon_cfg_csuctl` register supports external capture events 6 and 7:

- External capture event 6 is triggered using PMUSNAPSHOT interface.
- External capture event 7 is triggered as the *Arm® Architecture Reference Manual Supplement Memory System Resource Partitioning and Monitoring (MPAM)*, for Armv8-A describes.



Multiple capture events cannot be triggered within 32 cycles of each other.

4.4.5.5 MPAM error logging and reporting

CMN-700 implements programmable registers that can enable, disable, and modify MPAM error logging and reporting behavior.

The MPAM error status register, `cmn_hns_X_mpam_esr`, and the MPAM error control register, `cmn_hns_X_mpam_ecr`, define MPAM-related error logging.



The interconnect has two banks for these registers, S and NS, denoted by X in the register or register field name.

You can enable interrupt generation for MPAM-related errors by setting the `hnf_X_mpam_ecr_inten` field of the `cmn_hns_X_mpam_ecr` register to `0b1`. If interrupt generation is enabled, level-sensitive interrupts, `INTREQMPAMERRS`, `INTREQMPAMERRNS`, `INTREQMPAMERRRT`, or `INTREQMPAMERRRL`, are triggered for defined error cases.

The system can read `por_mpam_X_err_int_status`, `por_mpam_X_err_int_status_1` registers to determine the logical ID of the HN-F causing the interrupt.

MPAM error reporting has the following exceptions:



- When SLC size is OK, no errors are detected or reported.
- REQ PARTID or PMG out- of-range errors are not detected or reported when:
 - HN-F is in SFONLY mode.
 - MPAM features are disabled (by configuring the auxiliary control register).

For more information about MPAM errors, see *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:

- [5.2.3.15 cmn_hns_ns_mpam_ecr](#) on page 280
- [5.2.3.16 cmn_hns_ns_mpam_esr](#) on page 281
- [5.2.2.16 cmn_hns_s_mpam_ecr](#) on page 204
- [5.2.2.17 cmn_hns_s_mpam_esr](#) on page 205

4.4.5.6 Software-programmable MPAM override

CMN-700 has registers in various nodes to let software override the MPAM values generated by those nodes. The behavior of the override mechanism depends on the system configuration and security constraints.

The following nodes support a software override mechanism:

- RN-I
- RN-D
- CML-HA
- MXP

RN-Is, RN-Ds, and CML-HAs contain programmable registers to override the MPAM values for their own requests. The MXP contains programmable registers to override MPAM values for requests that RN-Fs generate.

The MXP software MPAM override mechanism is useful in the following situations:

- RN-F is a device that is compliant with CHI-D or CHI-E, and therefore supports MPAM, but MPAM is disabled in that device
- RN-F is a device that does not support MPAM

In these situations, the MXP MPAM override mechanism allows these devices to take part in the MPAM tracking scheme within the interconnect.

Each MXP contains an MPAM override register for each device port, `por_mxp_p0_mpam_override`, `por_mxp_p1_mpam_override`, `por_mxp_p2_mpam_override`, and `por_mxp_p3_mpam_override`.

Software can configure these registers to generate an MPAM override value for request flits that are uploaded on that device port by RN-Fs. These registers are applicable only when an RN-F is connected to the port.

For other device types, these registers are redundant and not used.

The following table shows the format of the override registers.

Table 4-22: `por_mxp_p{0,1,2,3}_mpam_override` register format

[63:25]	[24]	[23:17]	[16:8]	[7:5]	[4]	[3:1]	[0]
Reserved	REQ MPAM PMG	Reserved	REQ MPAM PartID	Reserved	REQ MPAM NS	Reserved	REQ MPAM override enable

The override enable bit indicates that the contents of the registers are valid, and therefore must be set when the other fields have the intended values.

The MXP MPAM override feature has the following security requirement:

- By default, the MPAM.NS subfield has the same value as the NS field of the REQ flit. When the override enable bit is set, the NS override field drives the NS field of the REQ flit.

The override behavior in the MXP depends on the CHI version that the RN-F supports and whether the enable bit is set as the following table shows.

Table 4-23: MXP override behavior

CHI_MPAM_ENABLE setting	RN-F CHI version	Override enable bit set or not set	Override behavior
1	CHI-B or CHI-C	Not set	All MPAM fields in the REQ flit, except the MPAM NS field, are driven to the default value, which is 0x00
1	CHI-B or CHI-C	Set	All fields, including the MPAM NS field, are overridden
1	CHI-D	Not set	The request uses the original MPAM values of the REQ flit
1	CHI-D	Set	All MPAM fields are overridden with the contents of the register



When the RN-F CHI version is CHI-E, it is not necessary for MPAM NS and Request NS to match. For example, you can use Request NS = 0 and MPAM NS = 1.

4.4.6 MTE support in HN-F

CMN-700 HN-F supports CHI-E *Memory Tagging Extensions* (MTE). For all coherent and non-coherent operations, if requests come with MTE opcodes, HN-F services them by fetching the required tags from SN-F.

In accordance with the CHI-E architecture, HN-F also flushes the dirty tags downstream to SN-F when required.

HN-F also supports finishing match operation in HN-F itself. By programming the `hns_mte_no_sn_match` in `hns_cfg_ctl` register, the match operation would always be done in HN-F and not sent downstream to SN-F.

HN-F supports to propagate clean the MTE tag to SN when the cacheline is victimized from SLC, by programming `slc_victim_clean_tag_transfer_en` to 1 in `cmn_hns_cfg_ctl` register.

To disable the MTE handling in HN-F when the software is running without MTE opcodes, program the `hnf_mte_mode_dis` bit of the `cmn_hns_cfg_ctl` register to 1. In this mode, HN-F ignores any requests for tags and drops dirty tags. For MTE Match requests, HN-F synthesizes a dummy Tag Match response to complete the protocol flow.



MTE is not supported for address regions belonging to CXL attached memory

4.5 HN-F class-based resource allocation and arbitration

CMN-700 POCQ supports class-based resource allocation and arbitration for increased flexibility and control.

4.5.1 Class assignment

QoS decoding happens inside the HN-F.

A request from RN is assigned a class based on QoS or Request Type (Opcode) of a dynamic, not previously retired, request. The `cmn_hns_class_ctl` config register provides configurability to

determine this class assignment. The assigned class is then used for POCQ resource allocation and arbitration to SLC/SF access. It is also used when arbitrating for TXREQ to an SN.

QoS based class assignment

`cmn_hns_pocq_qos_class_ctl` provides configurability to assign a class based on QoS min/max values.



All QoS values must be assigned one, and only one, class.

Request Type (Opcode) based class assignment

Incoming request opcode can be classified as follows:

Class 0

Read type requests

Class 1

Copy Back requests, including copyback wr+cmo

Class 2

Non Copy Back requests, including non-copyback wr+cmo

Class 3

All other requests

4.5.2 POCQ resource allocation

POCQ entries are allocated based on three configurable registers per class. These registers define different limits for each class.

`cmn_hns_pocq_alloc_class_dedicated`

Dedicated entries are reserved entries for a given class in POCQ. For a given class, if the number of requests in POCQ are less than the dedicated entries, then any incoming request to that class is not retried. Unused dedicated entries belonging to a class are unavailable for other classes to use.

`cmn_hns_pocq_alloc_class_max_allowed`

`max_allowed` indicates the maximum number of entries allowed for any given class in POCQ. When `max_allowed` is reached for a given class, any incoming request for that class is retried, even if POCQ has available entries.

`cmn_hns_pocq_alloc_class_contended_min`

`contended_min` indicates the required number of entries for a class under contention. A class is under contention when POCQ is full and requests are retried. These entries are not reserved. Instead, requests for a class below its contended minimum are given higher priority than ones above its contended minimum for credit grant arbitration.

The contended minimum value for a class is used for prioritizing classes for static credit grants after requests have been retried. When static credits are available and all classes have met or exceeded their number of dedicated entries, the number of active entries in the POCQ for each class is compared against the class's `contended_min` value. Classes whose number of active POCQ entries are below their `contended_min` value are prioritized for static credit grants.



The total number of dedicated entries, including SEQ reserved for snoop filter evictions, must not exceed the number of POCQ entries.

For any given class, Arm recommends that:

- `max_allowed > contended_min > dedicated`



`max_allowed` must not be less than the dedicated entries.

Retried requests are arbitrated per class to provide static credit grant back to requesting agent by the following priority order, highest to lowest:

- Request classes below `dedicated` with round robin arbitration across classes
- Request classes below `contended_min` with round robin arbitration across classes
- Request classes below `max_allowed` in weighted round robin across classes

Weights for weighted round robin can be configured in "`cmn_hns_class_retry_weight_ctl`". Within a given class, static grants are allocated to different RNs based on round robin arbitration.

The following figures show possible POCQ resource allocation for various classes for `dedicated` and `max_allowed`. `Dedicated` is a minimum for a class and `max_allowed` is a maximum per class.

Figure 4-3: POCQ possible dedicated

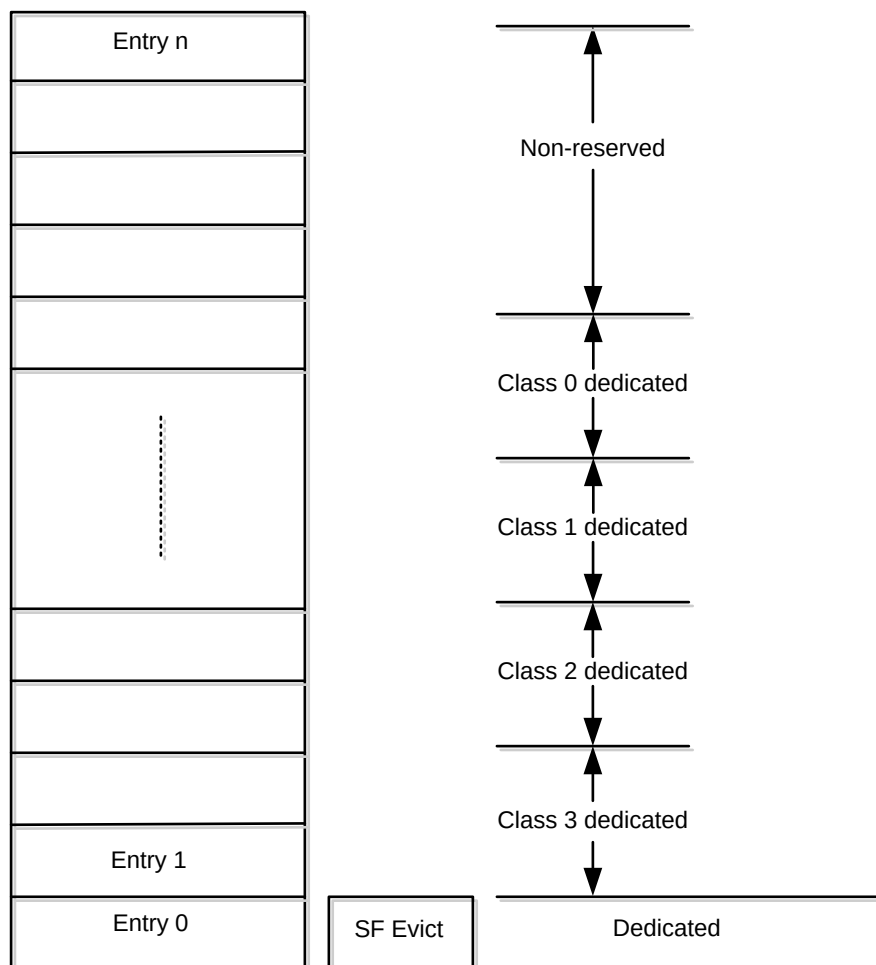
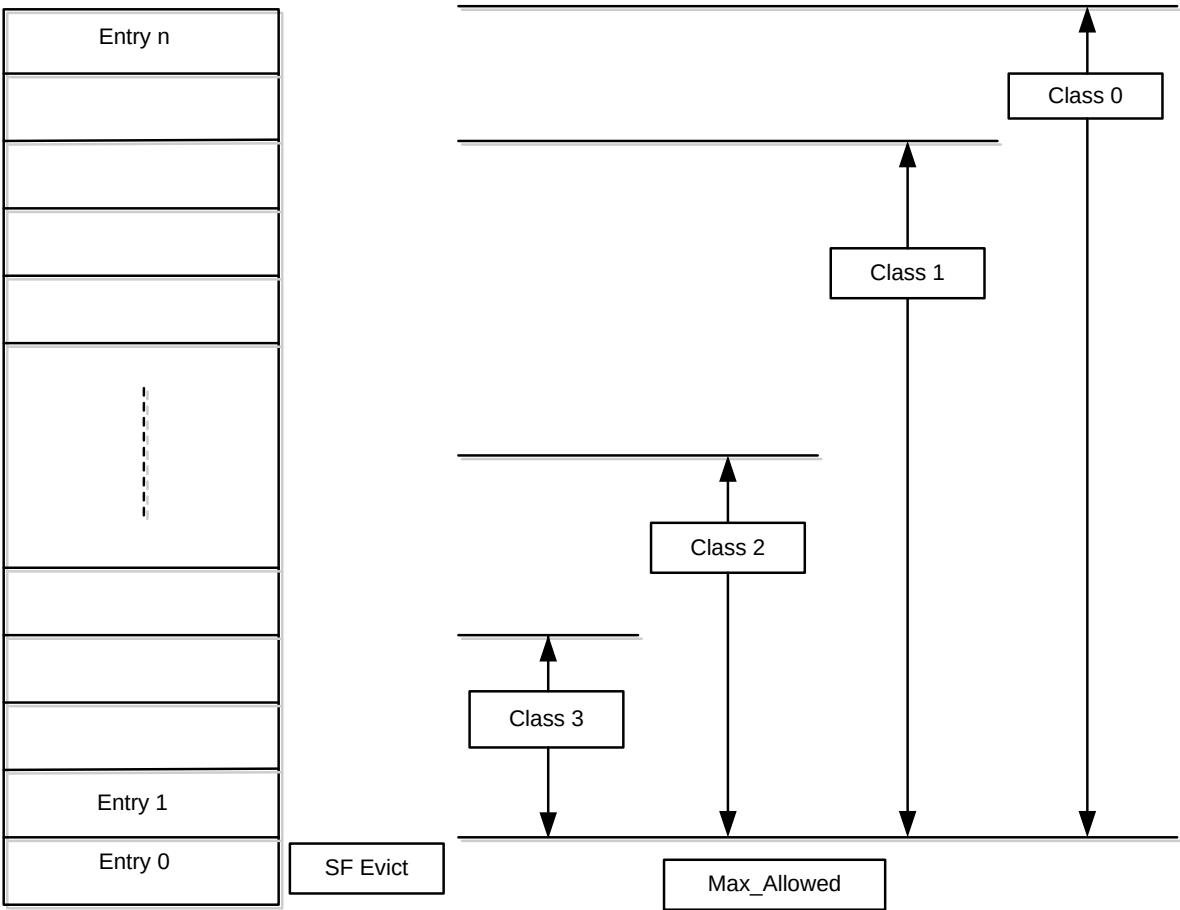


Figure 4-4: POCQ Default max_allocated



See [3.7.1.1 HN-F QoS support](#) on page 99 for the QoS class and POCQ resource availability

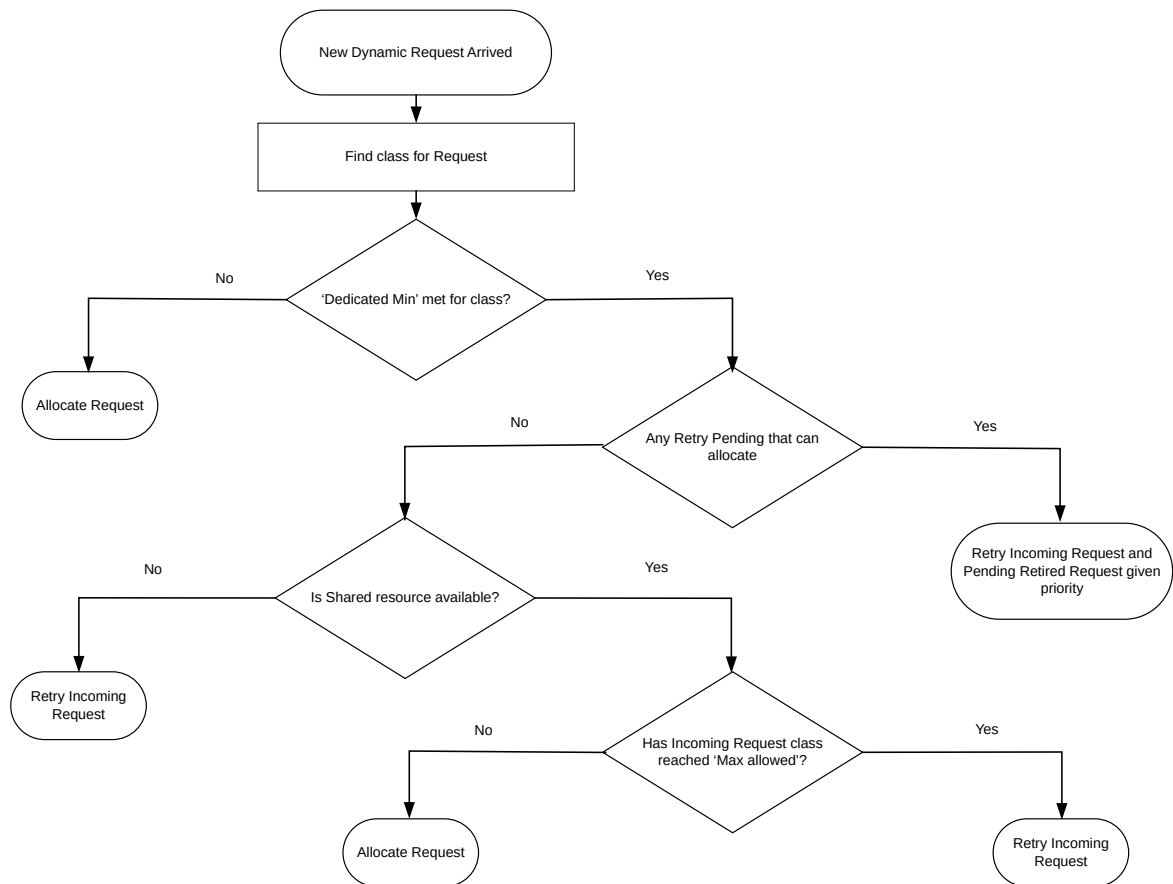
4.5.2.1 Request retry based on POCQ resource allocation

The following figure shows the decision making process to allocate an incoming dynamic request to POCQ or to retry.



An incoming static request cannot be retried so it is allocated in POCQ.

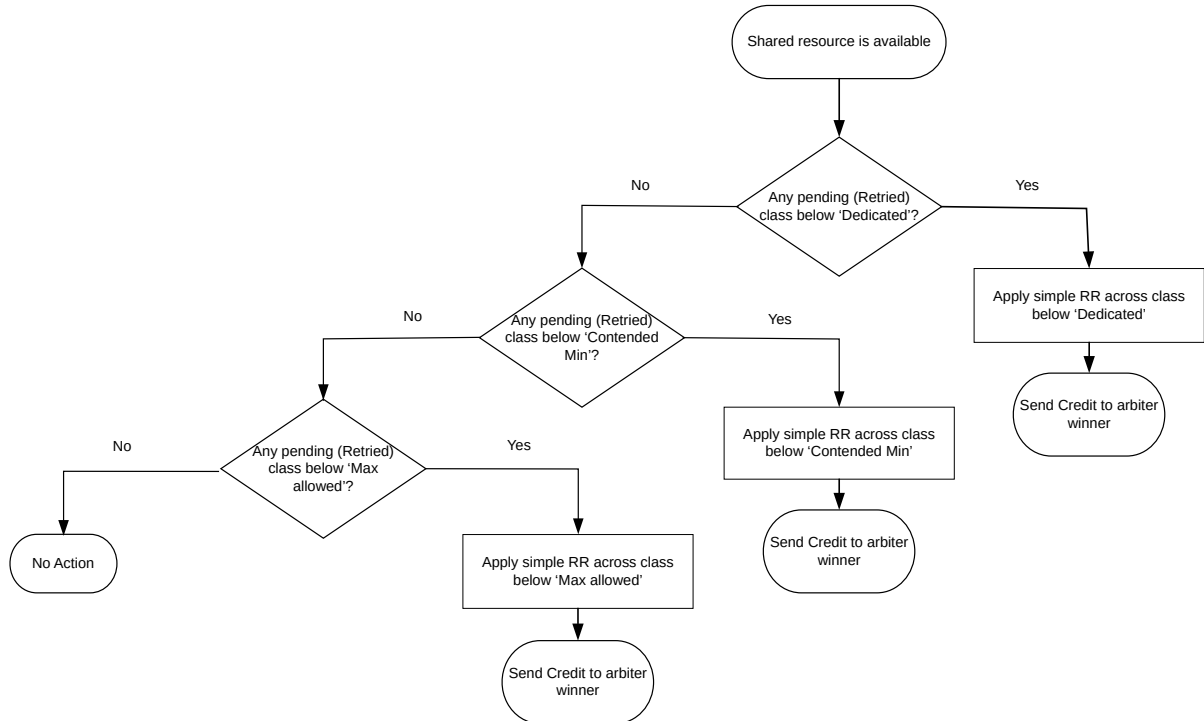
Figure 4-5: Request retry based on POCQ resource allocation



4.5.2.2 Credit grant for retried request

The following figure shows how credit is granted for different class requests when a request is retried. Within any given class, simple *Round Robin* (RR) arbitration is used.

Figure 4-6: Credit grant for retried request



4.5.3 POCQ request arbitration

Requests from POCQ to the SLC/SF pipe are selected by the following priority order, highest to lowest:

- Oldest requests in POCQ: ready for SLC/SF access
- Lookup requests: weighted RR per class
- Any eligible requests: weighted RR per class

Requests from POCQ to SN are selected by the following priority order, highest to lowest:

- Oldest requests in POCQ: ready to be sent to SN
- Static request: RR per class
- All pending SN requests: weighted RR per class

Configure the weights for weighted RR in `cmn_hns_class_pocq_arb_weight_ctl`. The same weights are used for SLC,SF and SN weighted RR.

5. Programmer's Model

This chapter describes the application-level registers and provides an overview for programming the CMN-700 interconnect.



This documentation details the *Super Home Node* (HN-S), which is only available through specific configurations and requires consent from Arm® to implement.



CMN-700 does not support a mixed configuration of HN-F and HN-S device types.



Any descriptions related to the *Fully coherent Home Node* (HN-F) also apply to the HN-S.

5.1 Register Summary

The register summary tables list the register used for HN-S

5.1.1 Configuration register summary

The following lists the configuration master registers used for HN-S

CRGM register summary

The following table shows the CFGM registers in offset order from the base memory

Table 5-1: CFGM register summary

Offset	Name	Type	Description
16'h900	por_info_global	RO	5.2.1.1 por_info_global on page 174
16'h908	por_info_global_1	RO	5.2.1.2 por_info_global_1 on page 176
16'h1900	por_ppu_int_enable	RW	5.2.1.3 por_ppu_int_enable on page 177
16'h1908	por_ppu_int_enable_1	RW	5.2.1.4 por_ppu_int_enable_1 on page 178

Offset	Name	Type	Description
16'h1910	por_ppu_int_status	W1C	5.2.1.5 por_ppu_int_status on page 179
16'h1918	por_ppu_int_status_1	W1C	5.2.1.6 por_ppu_int_status_1 on page 180
16'h1920	por_ppu_qactive_hyst	RW	5.2.1.7 por_ppu_qactive_hyst on page 181
16'h1928	por_mpam_s_err_int_status	W1C	5.2.1.8 por_mpam_s_err_int_status on page 182
16'h1930	por_mpam_s_err_int_status_1	W1C	5.2.1.9 por_mpam_s_err_int_status_1 on page 183
16'h1938	por_mpam_ns_err_int_status	W1C	5.2.1.10 por_mpam_ns_err_int_status on page 184
16'h1940	por_mpam_ns_err_int_status_1	W1C	5.2.1.11 por_mpam_ns_err_int_status_1 on page 185

5.1.2 HN-S MPAM_S register summary

This section lists the HN-S MPAM_S registers in CMN-700

HN-S MPAM_S register summary

The following table shows the HN-S MPAM_S registers in offset order from the base memory address.

Table 5-2: cmn_hns_mpam_s_cfg register summary

Offset	Name	Type	Description
16'h0	cmn_hns_mpam_s_node_info	RO	5.2.2.1 cmn_hns_mpam_s_node_info on page 186
16'h80	cmn_hns_mpam_s_child_info	RO	5.2.2.2 cmn_hns_mpam_s_child_info on page 187
16'h1000	cmn_hns_s_mpam_idr	RO	5.2.2.3 cmn_hns_s_mpam_idr on page 188
16'h1008	cmn_hns_mpam_sidr	RO	5.2.2.4 cmn_hns_mpam_sidr on page 190
16'h1018	cmn_hns_s_mpam_iidr	RO	5.2.2.5 cmn_hns_s_mpam_iidr on page 191
16'h1020	cmn_hns_s_mpam_aidr	RO	5.2.2.6 cmn_hns_s_mpam_aidr on page 192
16'h1028	cmn_hns_s_mpam_impl_idr	RO	5.2.2.7 cmn_hns_s_mpam_impl_idr on page 193
16'h1030	cmn_hns_s_mpam_cpor_idr	RO	5.2.2.8 cmn_hns_s_mpam_cpor_idr on page 194
16'h1038	cmn_hns_s_mpam_ccap_idr	RO	5.2.2.9 cmn_hns_s_mpam_ccap_idr on page 195
16'h1040	cmn_hns_s_mpam_mbw_idr	RO	5.2.2.10 cmn_hns_s_mpam_mbw_idr on page 196
16'h1048	cmn_hns_s_mpam_pri_idr	RO	5.2.2.11 cmn_hns_s_mpam_pri_idr on page 198
16'h1050	cmn_hns_s_mpam_partid_nrw_idr	RO	5.2.2.12 cmn_hns_s_mpam_partid_nrw_idr on page 199
16'h1080	cmn_hns_s_mpam_msmon_idr	RO	5.2.2.13 cmn_hns_s_mpam_msmon_idr on page 200
16'h1088	cmn_hns_s_mpam_csumon_idr	RO	5.2.2.14 cmn_hns_s_mpam_csumon_idr on page 201
16'h1090	cmn_hns_s_mpam_mbwumon_idr	RO	5.2.2.15 cmn_hns_s_mpam_mbwumon_idr on page 203

Offset	Name	Type	Description
16'h10F0	cmn_hns_s_mpam_ecr	RW	5.2.2.16 cmn_hns_s_mpam_ecr on page 204
16'h10F8	cmn_hns_s_mpam_esr	RW	5.2.2.17 cmn_hns_s_mpam_esr on page 205
16'h1100	cmn_hns_s_mpamcfg_part_sel	RW	5.2.2.18 cmn_hns_s_mpamcfg_part_sel on page 207
16'h1108	cmn_hns_s_mpamcfg_cmax	RW	5.2.2.19 cmn_hns_s_mpamcfg_cmax on page 208
16'h1200	cmn_hns_s_mpamcfg_mbw_min	RW	5.2.2.20 cmn_hns_s_mpamcfg_mbw_min on page 209
16'h1208	cmn_hns_s_mpamcfg_mbw_max	RW	5.2.2.21 cmn_hns_s_mpamcfg_mbw_max on page 210
16'h1220	cmn_hns_s_mpamcfg_mbw_winwd	RW	5.2.2.22 cmn_hns_s_mpamcfg_mbw_winwd on page 211
16'h1400	cmn_hns_s_mpamcfg_pri	RW	5.2.2.23 cmn_hns_s_mpamcfg_pri on page 212
16'h1500	cmn_hns_s_mpamcfg_mbw_prop	RW	5.2.2.24 cmn_hns_s_mpamcfg_mbw_prop on page 212
16'h1600	cmn_hns_s_mpamcfg_intpartid	RW	5.2.2.25 cmn_hns_s_mpamcfg_intpartid on page 213
16'h1800	cmn_hns_s_msmon_cfg_mon_sel	RW	5.2.2.26 cmn_hns_s_msmon_cfg_mon_sel on page 214
16'h1808	cmn_hns_s_msmon_capt_evnt	RW	5.2.2.27 cmn_hns_s_msmon_capt_evnt on page 215
16'h1810	cmn_hns_s_msmon_cfg_csuflt	RW	5.2.2.28 cmn_hns_s_msmon_cfg_csuflt on page 217
16'h1818	cmn_hns_s_msmon_cfg_csuctl	RW	5.2.2.29 cmn_hns_s_msmon_cfg_csuctl on page 218
16'h1820	cmn_hns_s_msmon_cfg_mbwuflt	RW	5.2.2.30 cmn_hns_s_msmon_cfg_mbwuflt on page 219
16'h1828	cmn_hns_s_msmon_cfg_mbwuctl	RW	5.2.2.31 cmn_hns_s_msmon_cfg_mbwuctl on page 221
16'h1840	cmn_hns_s_msmon_csu	RW	5.2.2.32 cmn_hns_s_msmon_csu on page 222
16'h1848	cmn_hns_s_msmon_csu_capture	RW	5.2.2.33 cmn_hns_s_msmon_csu_capture on page 223
16'h1860	cmn_hns_s_msmon_mbwu	RW	5.2.2.34 cmn_hns_s_msmon_mbwu on page 224
16'h1868	cmn_hns_s_msmon_mbwu_capture	RW	5.2.2.35 cmn_hns_s_msmon_mbwu_capture on page 225
16'h2000	cmn_hns_s_mpamcfg_cpbn	RW	5.2.2.36 cmn_hns_s_mpamcfg_cpbn on page 226
16'h8000	cmn_hns_rt_mpam_idr	RO	5.2.2.37 cmn_hns_rt_mpam_idr on page 227
16'h8028	cmn_hns_rt_mpam_impl_idr	RO	5.2.2.38 cmn_hns_rt_mpam_impl_idr on page 229
16'h8030	cmn_hns_rt_mpam_cpor_idr	RO	5.2.2.39 cmn_hns_rt_mpam_cpor_idr on page 230
16'h8038	cmn_hns_rt_mpam_ccap_idr	RO	5.2.2.40 cmn_hns_rt_mpam_ccap_idr on page 231
16'h8040	cmn_hns_rt_mpam_mbw_idr	RO	5.2.2.41 cmn_hns_rt_mpam_mbw_idr on page 232
16'h8048	cmn_hns_rt_mpam_pri_idr	RO	5.2.2.42 cmn_hns_rt_mpam_pri_idr on page 234
16'h8050	cmn_hns_rt_mpam_partid_nrw_idr	RO	5.2.2.43 cmn_hns_rt_mpam_partid_nrw_idr on page 235
16'h8080	cmn_hns_rt_mpam_msmon_idr	RO	5.2.2.44 cmn_hns_rt_mpam_msmon_idr on page 236
16'h8088	cmn_hns_rt_mpam_csumon_idr	RO	5.2.2.45 cmn_hns_rt_mpam_csumon_idr on page 237
16'h8090	cmn_hns_rt_mpam_mbwumon_idr	RO	5.2.2.46 cmn_hns_rt_mpam_mbwumon_idr on page 239
16'h80F0	cmn_hns_rt_mpam_ecr	RW	5.2.2.47 cmn_hns_rt_mpam_ecr on page 240
16'h80F8	cmn_hns_rt_mpam_esr	RW	5.2.2.48 cmn_hns_rt_mpam_esr on page 241
16'h8100	cmn_hns_rt_mpamcfg_part_sel	RW	5.2.2.49 cmn_hns_rt_mpamcfg_part_sel on page 243
16'h8108	cmn_hns_rt_mpamcfg_cmax	RW	5.2.2.50 cmn_hns_rt_mpamcfg_cmax on page 244
16'h8200	cmn_hns_rt_mpamcfg_mbw_min	RW	5.2.2.51 cmn_hns_rt_mpamcfg_mbw_min on page 244
16'h8208	cmn_hns_rt_mpamcfg_mbw_max	RW	5.2.2.52 cmn_hns_rt_mpamcfg_mbw_max on page 245
16'h8220	cmn_hns_rt_mpamcfg_mbw_winwd	RW	5.2.2.53 cmn_hns_rt_mpamcfg_mbw_winwd on page 247
16'h8400	cmn_hns_rt_mpamcfg_pri	RW	5.2.2.54 cmn_hns_rt_mpamcfg_pri on page 247
16'h8500	cmn_hns_rt_mpamcfg_mbw_prop	RW	5.2.2.55 cmn_hns_rt_mpamcfg_mbw_prop on page 248

Offset	Name	Type	Description
16'h8600	cmn_hns_rt_mpamcfg_intpartid	RW	5.2.2.56 cmn_hns_rt_mpamcfg_intpartid on page 249
16'h8800	cmn_hns_rt_msmon_cfg_mon_sel	RW	5.2.2.57 cmn_hns_rt_msmon_cfg_mon_sel on page 250
16'h8808	cmn_hns_rt_msmon_capt_evnt	RW	5.2.2.58 cmn_hns_rt_msmon_capt_evnt on page 251
16'h8810	cmn_hns_rt_msmon_cfg_csuflt	RW	5.2.2.59 cmn_hns_rt_msmon_cfg_csuflt on page 253
16'h8818	cmn_hns_rt_msmon_cfg_csuctl	RW	5.2.2.60 cmn_hns_rt_msmon_cfg_csuctl on page 254
16'h8820	cmn_hns_rt_msmon_cfg_mbwuflt	RW	5.2.2.61 cmn_hns_rt_msmon_cfg_mbwuflt on page 255
16'h8828	cmn_hns_rt_msmon_cfg_mbwuctl	RW	5.2.2.62 cmn_hns_rt_msmon_cfg_mbwuctl on page 257
16'h8840	cmn_hns_rt_msmon_csu	RW	5.2.2.63 cmn_hns_rt_msmon_csu on page 258
16'h8848	cmn_hns_rt_msmon_csu_capture	RW	5.2.2.64 cmn_hns_rt_msmon_csu_capture on page 259
16'h8860	cmn_hns_rt_msmon_mbwu	RW	5.2.2.65 cmn_hns_rt_msmon_mbwu on page 260
16'h8868	cmn_hns_rt_msmon_mbwu_capture	RW	5.2.2.66 cmn_hns_rt_msmon_mbwu_capture on page 261
16'h9000	cmn_hns_rt_mpamcfg_cpbn	RW	5.2.2.67 cmn_hns_rt_mpamcfg_cpbn on page 262

5.1.3 HN-S MPAM_NS register summary

This section lists the HN-S MPAM_NS registers in CMN-700

HN-S MPAM_NS register summary

The following table shows the HN-NS MPAM_S registers in offset order from the base memory address.

Table 5-3: cmn_hns_mpam_ns_cfg register summary

Offset	Name	Type	Description
16'h0	cmn_hns_mpam_ns_node_info	RO	5.2.3.1 cmn_hns_mpam_ns_node_info on page 263
16'h80	cmn_hns_mpam_ns_child_info	RO	5.2.3.2 cmn_hns_mpam_ns_child_info on page 264
16'h1000	cmn_hns_ns_mpam_idr	RO	5.2.3.3 cmn_hns_ns_mpam_idr on page 265
16'h1018	cmn_hns_mpam_iidr	RO	5.2.3.4 cmn_hns_mpam_iidr on page 267
16'h1020	cmn_hns_mpam_aidr	RO	5.2.3.5 cmn_hns_mpam_aidr on page 268
16'h1028	cmn_hns_ns_mpam_impl_idr	RO	5.2.3.6 cmn_hns_ns_mpam_impl_idr on page 269
16'h1030	cmn_hns_ns_mpam_cpor_idr	RO	5.2.3.7 cmn_hns_ns_mpam_cpor_idr on page 270
16'h1038	cmn_hns_ns_mpam_ccap_idr	RO	5.2.3.8 cmn_hns_ns_mpam_ccap_idr on page 271
16'h1040	cmn_hns_ns_mpam_mbw_idr	RO	5.2.3.9 cmn_hns_ns_mpam_mbw_idr on page 272
16'h1048	cmn_hns_ns_mpam_pri_idr	RO	5.2.3.10 cmn_hns_ns_mpam_pri_idr on page 273
16'h1050	cmn_hns_ns_mpam_partid_nrw_idr	RO	5.2.3.11 cmn_hns_ns_mpam_partid_nrw_idr on page 275
16'h1080	cmn_hns_ns_mpam_msmon_idr	RO	5.2.3.12 cmn_hns_ns_mpam_msmon_idr on page 276
16'h1088	cmn_hns_ns_mpam_csumon_idr	RO	5.2.3.13 cmn_hns_ns_mpam_csumon_idr on page 277
16'h1090	cmn_hns_ns_mpam_mbwumon_idr	RO	5.2.3.14 cmn_hns_ns_mpam_mbwumon_idr on page 279
16'h10F0	cmn_hns_ns_mpam_ecr	RW	5.2.3.15 cmn_hns_ns_mpam_ecr on page 280
16'h10F8	cmn_hns_ns_mpam_esr	RW	5.2.3.16 cmn_hns_ns_mpam_esr on page 281
16'h1100	cmn_hns_ns_mpamcfg_part_sel	RW	5.2.3.17 cmn_hns_ns_mpamcfg_part_sel on page 283

Offset	Name	Type	Description
16'h1108	cmn_hns_ns_mpamcfg_cmax	RW	5.2.3.18 cmn_hns_ns_mpamcfg_cmax on page 284
16'h1200	cmn_hns_ns_mpamcfg_mbw_min	RW	5.2.3.19 cmn_hns_ns_mpamcfg_mbw_min on page 285
16'h1208	cmn_hns_ns_mpamcfg_mbw_max	RW	5.2.3.20 cmn_hns_ns_mpamcfg_mbw_max on page 286
16'h1220	cmn_hns_ns_mpamcfg_mbw_winwd	RW	5.2.3.21 cmn_hns_ns_mpamcfg_mbw_winwd on page 287
16'h1400	cmn_hns_ns_mpamcfg_pri	RW	5.2.3.22 cmn_hns_ns_mpamcfg_pri on page 288
16'h1500	cmn_hns_ns_mpamcfg_mbw_prop	RW	5.2.3.23 cmn_hns_ns_mpamcfg_mbw_prop on page 288
16'h1600	cmn_hns_ns_mpamcfg_intpartid	RW	5.2.3.24 cmn_hns_ns_mpamcfg_intpartid on page 289
16'h1800	cmn_hns_ns_msmon_cfg_mon_sel	RW	5.2.3.25 cmn_hns_ns_msmon_cfg_mon_sel on page 290
16'h1808	cmn_hns_ns_msmon_capt_evnt	RW	5.2.3.26 cmn_hns_ns_msmon_capt_evnt on page 291
16'h1810	cmn_hns_ns_msmon_cfg_csuflt	RW	5.2.3.27 cmn_hns_ns_msmon_cfg_csuflt on page 293
16'h1818	cmn_hns_ns_msmon_cfg_csuctl	RW	5.2.3.28 cmn_hns_ns_msmon_cfg_csuctl on page 294
16'h1820	cmn_hns_ns_msmon_cfg_mbwuflt	RW	5.2.3.29 cmn_hns_ns_msmon_cfg_mbwuflt on page 296
16'h1828	cmn_hns_ns_msmon_cfg_mbwuctl	RW	5.2.3.30 cmn_hns_ns_msmon_cfg_mbwuctl on page 297
16'h1840	cmn_hns_ns_msmon_csu	RW	5.2.3.31 cmn_hns_ns_msmon_csu on page 299
16'h1848	cmn_hns_ns_msmon_csu_capture	RW	5.2.3.32 cmn_hns_ns_msmon_csu_capture on page 300
16'h1860	cmn_hns_ns_msmon_mbwu	RW	5.2.3.33 cmn_hns_ns_msmon_mbwu on page 301
16'h1868	cmn_hns_ns_msmon_mbwu_capture	RW	5.2.3.34 cmn_hns_ns_msmon_mbwu_capture on page 302
16'h2000	cmn_hns_ns_mpamcfg_cpbn	RW	5.2.3.35 cmn_hns_ns_mpamcfg_cpbn on page 303
16'h8000	cmn_hns_rl_mpam_idr	RO	5.2.3.36 cmn_hns_rl_mpam_idr on page 304
16'h8028	cmn_hns_rl_mpam_impl_idr	RO	5.2.3.37 cmn_hns_rl_mpam_impl_idr on page 306
16'h8030	cmn_hns_rl_mpam_cpor_idr	RO	5.2.3.38 cmn_hns_rl_mpam_cpor_idr on page 307
16'h8038	cmn_hns_rl_mpam_ccap_idr	RO	5.2.3.39 cmn_hns_rl_mpam_ccap_idr on page 307
16'h8040	cmn_hns_rl_mpam_mbw_idr	RO	5.2.3.40 cmn_hns_rl_mpam_mbw_idr on page 309
16'h8048	cmn_hns_rl_mpam_pri_idr	RO	5.2.3.41 cmn_hns_rl_mpam_pri_idr on page 310
16'h8050	cmn_hns_rl_mpam_partid_nrw_idr	RO	5.2.3.42 cmn_hns_rl_mpam_partid_nrw_idr on page 312
16'h8080	cmn_hns_rl_mpam_msmon_idr	RO	5.2.3.43 cmn_hns_rl_mpam_msmon_idr on page 313
16'h8088	cmn_hns_rl_mpam_csumon_idr	RO	5.2.3.44 cmn_hns_rl_mpam_csumon_idr on page 314
16'h8090	cmn_hns_rl_mpam_mbwumon_idr	RO	5.2.3.45 cmn_hns_rl_mpam_mbwumon_idr on page 316
16'h80F0	cmn_hns_rl_mpam_ecr	RW	5.2.3.46 cmn_hns_rl_mpam_ecr on page 317
16'h80F8	cmn_hns_rl_mpam_esr	RW	5.2.3.47 cmn_hns_rl_mpam_esr on page 318
16'h8100	cmn_hns_rl_mpamcfg_part_sel	RW	5.2.3.48 cmn_hns_rl_mpamcfg_part_sel on page 320
16'h8108	cmn_hns_rl_mpamcfg_cmax	RW	5.2.3.49 cmn_hns_rl_mpamcfg_cmax on page 321
16'h8200	cmn_hns_rl_mpamcfg_mbw_min	RW	5.2.3.50 cmn_hns_rl_mpamcfg_mbw_min on page 322
16'h8208	cmn_hns_rl_mpamcfg_mbw_max	RW	5.2.3.51 cmn_hns_rl_mpamcfg_mbw_max on page 323
16'h8220	cmn_hns_rl_mpamcfg_mbw_winwd	RW	5.2.3.52 cmn_hns_rl_mpamcfg_mbw_winwd on page 324
16'h8400	cmn_hns_rl_mpamcfg_pri	RW	5.2.3.53 cmn_hns_rl_mpamcfg_pri on page 325
16'h8500	cmn_hns_rl_mpamcfg_mbw_prop	RW	5.2.3.54 cmn_hns_rl_mpamcfg_mbw_prop on page 325
16'h8600	cmn_hns_rl_mpamcfg_intpartid	RW	5.2.3.55 cmn_hns_rl_mpamcfg_intpartid on page 326
16'h8800	cmn_hns_rl_msmon_cfg_mon_sel	RW	5.2.3.56 cmn_hns_rl_msmon_cfg_mon_sel on page 327
16'h8808	cmn_hns_rl_msmon_capt_evnt	RW	5.2.3.57 cmn_hns_rl_msmon_capt_evnt on page 328

Offset	Name	Type	Description
16'h8810	cmn_hns_rl_msmon_cfg_csuflt	RW	5.2.3.58 cmn_hns_rl_msmon_cfg_csuflt on page 330
16'h8818	cmn_hns_rl_msmon_cfg_csuctl	RW	5.2.3.59 cmn_hns_rl_msmon_cfg_csuctl on page 331
16'h8820	cmn_hns_rl_msmon_cfg_mbwuflt	RW	5.2.3.60 cmn_hns_rl_msmon_cfg_mbwuflt on page 332
16'h8828	cmn_hns_rl_msmon_cfg_mbwuctl	RW	5.2.3.61 cmn_hns_rl_msmon_cfg_mbwuctl on page 334
16'h8840	cmn_hns_rl_msmon_csu	RW	5.2.3.62 cmn_hns_rl_msmon_csu on page 335
16'h8848	cmn_hns_rl_msmon_csu_capture	RW	5.2.3.63 cmn_hns_rl_msmon_csu_capture on page 336
16'h8860	cmn_hns_rl_msmon_mbwu	RW	5.2.3.64 cmn_hns_rl_msmon_mbwu on page 337
16'h8868	cmn_hns_rl_msmon_mbwu_capture	RW	5.2.3.65 cmn_hns_rl_msmon_mbwu_capture on page 338
16'h9000	cmn_hns_rl_mpamcfg_cpbm	RW	5.2.3.66 cmn_hns_rl_mpamcfg_cpbm on page 339

5.1.4 HN-S register summary

This section lists the HN-S registers in CMN-700

HN-S register summary

The following table shows the HN-S registers in offset order from the base memory address.

Table 5-4: cmn_hns_cfg register summary

Offset	Name	Type	Description
16'h0	cmn_hns_node_info	RO	5.2.4.1 cmn_hns_node_info on page 340
16'h80	cmn_hns_child_info	RO	5.2.4.2 cmn_hns_child_info on page 341
16'h980	cmn_hns_scr	RW	5.2.4.3 cmn_hns_scr on page 342
16'h988	cmn_hns_rcr	RW	5.2.4.4 cmn_hns_rcr on page 343
16'h900	cmn_hns_unit_info	RO	5.2.4.5 cmn_hns_unit_info on page 345
16'h908	cmn_hns_unit_info_1	RO	5.2.4.6 cmn_hns_unit_info_1 on page 346
16'hA00	cmn_hns_cfg_ctl	RW	5.2.4.7 cmn_hns_cfg_ctl on page 347
16'hA08	cmn_hns_aux_ctl	RW	5.2.4.8 cmn_hns_aux_ctl on page 352
16'hA10	cmn_hns_aux_ctl_1	RW	5.2.4.9 cmn_hns_aux_ctl_1 on page 354
16'hA18	cmn_hns_cbusy_limit_ctl	RW	5.2.4.10 cmn_hns_cbusy_limit_ctl on page 356
16'hA20	cmn_hns_txrsp_arb_weight_ctl	RW	5.2.4.11 cmn_hns_txrsp_arb_weight_ctl on page 357
16'hA28	cmn_hns_cbusy_mode_ctl	RW	5.2.4.12 cmn_hns_cbusy_mode_ctl on page 358
16'hA30	cmn_hns_lbt_cfg_ctl	RW	5.2.4.13 cmn_hns_lbt_cfg_ctl on page 360
16'hA38	cmn_hns_lbt_aux_ctl	RW	5.2.4.14 cmn_hns_lbt_aux_ctl on page 361
16'h1900	cmn_hns_ppu_pwpr	RW	5.2.4.15 cmn_hns_ppu_pwpr on page 363
16'h1908	cmn_hns_ppu_pwsr	RO	5.2.4.16 cmn_hns_ppu_pwsr on page 364
16'h1914	cmn_hns_ppu_misr	RO	5.2.4.17 cmn_hns_ppu_misr on page 365
16'h28B0	cmn_hns_ppu_idr0	RO	5.2.4.18 cmn_hns_ppu_idr0 on page 366
16'h28B4	cmn_hns_ppu_idr1	RO	5.2.4.19 cmn_hns_ppu_idr1 on page 368
16'h28C8	cmn_hns_ppu_iidr	RO	5.2.4.20 cmn_hns_ppu_iidr on page 369
16'h28CC	cmn_hns_ppu_aidr	RO	5.2.4.21 cmn_hns_ppu_aidr on page 370

Offset	Name	Type	Description
16'h1A00	cmn_hns_ppu_dyn_ret_threshold	RW	5.2.4.22 cmn_hns_ppu_dyn_ret_threshold on page 370
16'hA80	cmn_hns_qos_band	RO	5.2.4.23 cmn_hns_qos_band on page 372
16'hE000	cmn_hns_errfr	RO	5.2.4.24 cmn_hns_errfr on page 373
16'hE008	cmn_hns_errctlr	RW	5.2.4.25 cmn_hns_errctlr on page 375
16'hE010	cmn_hns_errstatus	W1C	5.2.4.26 cmn_hns_errstatus on page 376
16'hE018	cmn_hns_erraddr	RW	5.2.4.27 cmn_hns_erraddr on page 378
16'hE020	cmn_hns_errmisc0	RW	5.2.4.28 cmn_hns_errmisc0 on page 379
16'hE028	cmn_hns_errmisc1	RW	5.2.4.29 cmn_hns_errmisc1 on page 380
16'hE800	cmn_hns_errpfgf	RO	5.2.4.30 cmn_hns_errpfgf on page 382
16'hE808	cmn_hns_errpfgctl	RW	5.2.4.31 cmn_hns_errpfgctl on page 383
16'hE810	cmn_hns_errpfgcdn	RW	5.2.4.32 cmn_hns_errpfgcdn on page 385
16'hE040	cmn_hns_errfr_NS	RO	5.2.4.33 cmn_hns_errfr_NS on page 386
16'hE048	cmn_hns_errctlr_NS	RW	5.2.4.34 cmn_hns_errctlr_NS on page 387
16'hE050	cmn_hns_errstatus_NS	W1C	5.2.4.35 cmn_hns_errstatus_NS on page 389
16'hE058	cmn_hns_erraddr_NS	RW	5.2.4.36 cmn_hns_erraddr_NS on page 390
16'hE060	cmn_hns_errmisc0_NS	RW	5.2.4.37 cmn_hns_errmisc0_NS on page 391
16'hE068	cmn_hns_errmisc1_NS	RW	5.2.4.38 cmn_hns_errmisc1_NS on page 392
16'hE840	cmn_hns_errpfgf_NS	RO	5.2.4.39 cmn_hns_errpfgf_NS on page 393
16'hE848	cmn_hns_errpfgctl_NS	RW	5.2.4.40 cmn_hns_errpfgctl_NS on page 394
16'hE850	cmn_hns_errpfgcdn_NS	RW	5.2.4.41 cmn_hns_errpfgcdn_NS on page 396
16'hED00	cmn_hns_errcapctl	RW	5.2.4.42 cmn_hns_errcapctl on page 397
16'hEE00	cmn_hns_errgsr	RO	5.2.4.43 cmn_hns_errgsr on page 398
16'hEE10	cmn_hns_erridr	RO	5.2.4.44 cmn_hns_erridr on page 399
16'hEFA8	cmn_hns_errdevaff	RO	5.2.4.45 cmn_hns_errdevaff on page 400
16'hEFB8	cmn_hns_errdevarch	RO	5.2.4.46 cmn_hns_errdevarch on page 401
16'hEFC8	cmn_hns_errdevid	RO	5.2.4.47 cmn_hns_errdevid on page 402
16'hEFD0	cmn_hns_errpidr45	RO	5.2.4.48 cmn_hns_errpidr45 on page 403
16'hEFE0	cmn_hns_errpidr01	RO	5.2.4.49 cmn_hns_errpidr01 on page 404
16'hEFE8	cmn_hns_errpidr23	RO	5.2.4.50 cmn_hns_errpidr23 on page 404
16'hEFF0	cmn_hns_errcidr01	RO	5.2.4.51 cmn_hns_errcidr01 on page 405
16'hEFF8	cmn_hns_errcidr23	RO	5.2.4.52 cmn_hns_errcidr23 on page 406
16'hE030	cmn_hns_err_inj	RW	5.2.4.53 cmn_hns_err_inj on page 407
16'hE938	cmn_hns_byte_par_err_inj	WO	5.2.4.54 cmn_hns_byte_par_err_inj on page 408
16'hC00	cmn_hns_slc_lock_ways	RW	5.2.4.55 cmn_hns_slc_lock_ways on page 409
16'hC08	cmn_hns_slc_lock_base0	RW	5.2.4.56 cmn_hns_slc_lock_base0 on page 411
16'hC10	cmn_hns_slc_lock_base1	RW	5.2.4.57 cmn_hns_slc_lock_base1 on page 412
16'hC18	cmn_hns_slc_lock_base2	RW	5.2.4.58 cmn_hns_slc_lock_base2 on page 413
16'hC20	cmn_hns_slc_lock_base3	RW	5.2.4.59 cmn_hns_slc_lock_base3 on page 414
16'hC28	cmn_hns_rni_region_vec	RW	5.2.4.60 cmn_hns_rni_region_vec on page 415
16'hC30	cmn_hns_rnd_region_vec	RW	5.2.4.61 cmn_hns_rnd_region_vec on page 416

Offset	Name	Type	Description
16'hC38	cmn_hns_rnf_region_vec	RW	5.2.4.62 cmn_hns_rnf_region_vec on page 417
16'hC40	cmn_hns_rnf_region_vec1	RW	5.2.4.63 cmn_hns_rnf_region_vec1 on page 418
16'hC48	cmn_hns_slcway_partition0_rnf_vec	RW	5.2.4.64 cmn_hns_slcway_partition0_rnf_vec on page 420
16'hC50	cmn_hns_slcway_partition1_rnf_vec	RW	5.2.4.65 cmn_hns_slcway_partition1_rnf_vec on page 421
16'hC58	cmn_hns_slcway_partition2_rnf_vec	RW	5.2.4.66 cmn_hns_slcway_partition2_rnf_vec on page 422
16'hC60	cmn_hns_slcway_partition3_rnf_vec	RW	5.2.4.67 cmn_hns_slcway_partition3_rnf_vec on page 423
16'hCB0	cmn_hns_slcway_partition0_rnf_vec1	RW	5.2.4.68 cmn_hns_slcway_partition0_rnf_vec1 on page 424
16'hCB8	cmn_hns_slcway_partition1_rnf_vec1	RW	5.2.4.69 cmn_hns_slcway_partition1_rnf_vec1 on page 425
16'hCC0	cmn_hns_slcway_partition2_rnf_vec1	RW	5.2.4.70 cmn_hns_slcway_partition2_rnf_vec1 on page 426
16'hCC8	cmn_hns_slcway_partition3_rnf_vec1	RW	5.2.4.71 cmn_hns_slcway_partition3_rnf_vec1 on page 427
16'hC68	cmn_hns_slcway_partition0_rni_vec	RW	5.2.4.72 cmn_hns_slcway_partition0_rni_vec on page 428
16'hC70	cmn_hns_slcway_partition1_rni_vec	RW	5.2.4.73 cmn_hns_slcway_partition1_rni_vec on page 429
16'hC78	cmn_hns_slcway_partition2_rni_vec	RW	5.2.4.74 cmn_hns_slcway_partition2_rni_vec on page 430
16'hC80	cmn_hns_slcway_partition3_rni_vec	RW	5.2.4.75 cmn_hns_slcway_partition3_rni_vec on page 431
16'hC88	cmn_hns_slcway_partition0_rnd_vec	RW	5.2.4.76 cmn_hns_slcway_partition0_rnd_vec on page 432
16'hC90	cmn_hns_slcway_partition1_rnd_vec	RW	5.2.4.77 cmn_hns_slcway_partition1_rnd_vec on page 433
16'hC98	cmn_hns_slcway_partition2_rnd_vec	RW	5.2.4.78 cmn_hns_slcway_partition2_rnd_vec on page 434
16'hCA0	cmn_hns_slcway_partition3_rnd_vec	RW	5.2.4.79 cmn_hns_slcway_partition3_rnd_vec on page 435
16'hCA8	cmn_hns_rn_region_lock	RW	5.2.4.80 cmn_hns_rn_region_lock on page 436
16'hCD0	cmn_hns_sf_cxg_blocked_ways	RW	5.2.4.81 cmn_hns_sf_cxg_blocked_ways on page 437
16'hCE0	cmn_hns_cxg_ha_metadata_exclusion_list	RW	5.2.4.82 cmn_hns_cxg_ha_metadata_exclusion_list on page 439
16'hCD8	cmn_hns_cxg_ha_smp_exclusion_list	RW	5.2.4.83 cmn_hns_cxg_ha_smp_exclusion_list on page 440
16'hCF0	hn_sam_hash_addr_mask_reg	RW	5.2.4.84 hn_sam_hash_addr_mask_reg on page 441
16'hCF8	hn_sam_region_cmp_addr_mask_reg	RW	5.2.4.85 hn_sam_region_cmp_addr_mask_reg on page 442
16'hD48	cmn_hns_sam_cfg1_def_hashed_region	RW	5.2.4.86 cmn_hns_sam_cfg1_def_hashed_region on page 443

Offset	Name	Type	Description
16'hD50	cmn_hns_sam_cfg2_def_hashed_region	RW	5.2.4.87 cmn_hns_sam_cfg2_def_hashed_region on page 444
16'hD00	cmn_hns_sam_control	RW	5.2.4.88 cmn_hns_sam_control on page 445
16'hD28	cmn_hns_sam_control2	RW	5.2.4.89 cmn_hns_sam_control2 on page 447
16'hD08	cmn_hns_sam_memregion0	RW	5.2.4.90 cmn_hns_sam_memregion0 on page 448
16'hD38	cmn_hns_sam_memregion0_end_addr	RW	5.2.4.91 cmn_hns_sam_memregion0_end_addr on page 449
16'hD10	cmn_hns_sam_memregion1	RW	5.2.4.92 cmn_hns_sam_memregion1 on page 450
16'hD40	cmn_hns_sam_memregion1_end_addr	RW	5.2.4.93 cmn_hns_sam_memregion1_end_addr on page 452
16'hD18	cmn_hns_sam_sn_properties	RW	5.2.4.94 cmn_hns_sam_sn_properties on page 453
16'hD20	cmn_hns_sam_6sn_nodeid	RW	5.2.4.95 cmn_hns_sam_6sn_nodeid on page 456
16'hCE8	cmn_hns_sam_sn_properties1	RW	5.2.4.96 cmn_hns_sam_sn_properties1 on page 457
16'hD30	cmn_hns_sam_sn_properties2	RW	5.2.4.97 cmn_hns_sam_sn_properties2 on page 459
{0-4} 16'hF80 + #{8 * index} {5-31} 16'h6000 + #{8 * index}	cmn_hns_cml_port_aggr_grp0-4_add_mask	RW	5.2.4.98 cmn_hns_cml_port_aggr_grp0-4_add_mask on page 462
{0-4} 16'hF80 + #{8 * index} {5-31} 16'hF80 + #{8 * index}	cmn_hns_cml_port_aggr_grp5-31_add_mask	RW	5.2.4.99 cmn_hns_cml_port_aggr_grp5-31_add_mask on page 463
{0-1} 16'hFB0 : 16'hFB8 {2-12} 16'h6110 : 16'h6160	cmn_hns_cml_port_aggr_grp_reg0-12	RW	5.2.4.100 cmn_hns_cml_port_aggr_grp_reg0-12 on page 464
16'hFD0	cmn_hns_cml_port_aggr_ctrl_reg	RW	5.2.4.101 cmn_hns_cml_port_aggr_ctrl_reg on page 465
{1-6} 16'h6208 : 16'h6230	cmn_hns_cml_port_aggr_ctrl_reg1-6	RW	5.2.4.102 cmn_hns_cml_port_aggr_ctrl_reg1-6 on page 469
16'hF50	cmn_hns_abf_lo_addr	RW	5.2.4.103 cmn_hns_abf_lo_addr on page 473
16'hF58	cmn_hns_abf_hi_addr	RW	5.2.4.104 cmn_hns_abf_hi_addr on page 474
16'hF60	cmn_hns_abf_pr	RW	5.2.4.105 cmn_hns_abf_pr on page 475
16'hF68	cmn_hns_abf_sr	RO	5.2.4.106 cmn_hns_abf_sr on page 476
16'h1000	cmn_hns_cbusy_write_limit_ctl	RW	5.2.4.107 cmn_hns_cbusy_write_limit_ctl on page 478
16'h1008	cmn_hns_cbusy_resp_ctl	RW	5.2.4.108 cmn_hns_cbusy_resp_ctl on page 479
16'h1010	cmn_hns_cbusy_sn_ctl	RW	5.2.4.109 cmn_hns_cbusy_sn_ctl on page 481
16'h1018	cmn_hns_lbt_cbusy_ctl	RW	5.2.4.110 cmn_hns_lbt_cbusy_ctl on page 482
16'h1020	cmn_hns_pocq_alloc_class_dedicated	RW	5.2.4.111 cmn_hns_pocq_alloc_class_dedicated on page 483
16'h1028	cmn_hns_pocq_alloc_class_max_allowed	RW	5.2.4.112 cmn_hns_pocq_alloc_class_max_allowed on page 484

Offset	Name	Type	Description
16'h1030	cmn_hns_pocq_alloc_class_contended_min	RW	5.2.4.113 cmn_hns_pocq_alloc_class_contended_min on page 486
16'h1038	cmn_hns_pocq_alloc_misc_max_allowed	RW	5.2.4.114 cmn_hns_pocq_alloc_misc_max_allowed on page 487
16'h1040	cmn_hns_class_ctl	RW	5.2.4.115 cmn_hns_class_ctl on page 488
16'h1048	cmn_hns_pocq_qos_class_ctl	RW	5.2.4.116 cmn_hns_pocq_qos_class_ctl on page 489
16'h1050	cmn_hns_class_pocq_arb_weight_ctl	RW	5.2.4.117 cmn_hns_class_pocq_arb_weight_ctl on page 490
16'h1058	cmn_hns_class_retry_weight_ctl	RW	5.2.4.118 cmn_hns_class_retry_weight_ctl on page 492
16'h1060	cmn_hns_pocq_misc_retry_weight_ctl	RW	5.2.4.119 cmn_hns_pocq_misc_retry_weight_ctl on page 493
16'hFE0	cmn_hns_partner_scratch_reg0	RW	5.2.4.120 cmn_hns_partner_scratch_reg0 on page 495
16'hFE8	cmn_hns_partner_scratch_reg1	RW	5.2.4.121 cmn_hns_partner_scratch_reg1 on page 496
16'hB80	cmn_hns_cfg_slcsf_dbgrd	WO	5.2.4.122 cmn_hns_cfg_slcsf_dbgrd on page 497
16'hB88	cmn_hns_slc_cache_access_slc_tag	RO	5.2.4.123 cmn_hns_slc_cache_access_slc_tag on page 498
16'hB90	cmn_hns_slc_cache_access_slc_tag1	RO	5.2.4.124 cmn_hns_slc_cache_access_slc_tag1 on page 499
16'hB98	cmn_hns_slc_cache_access_slc_data	RO	5.2.4.125 cmn_hns_slc_cache_access_slc_data on page 500
16'hBC0	cmn_hns_slc_cache_access_slc_mte_tag	RO	5.2.4.126 cmn_hns_slc_cache_access_slc_mte_tag on page 501
16'hBA0	cmn_hns_slc_cache_access_sf_tag	RO	5.2.4.127 cmn_hns_slc_cache_access_sf_tag on page 502
16'hBA8	cmn_hns_slc_cache_access_sf_tag1	RO	5.2.4.128 cmn_hns_slc_cache_access_sf_tag1 on page 503
16'hBB0	cmn_hns_slc_cache_access_sf_tag2	RO	5.2.4.129 cmn_hns_slc_cache_access_sf_tag2 on page 504
16'hD900	cmn_hns_pmu_event_sel	RW	5.2.4.130 cmn_hns_pmu_event_sel on page 505
16'hD908	cmn_hns_pmu_mpam_sel	RW	5.2.4.131 cmn_hns_pmu_mpam_sel on page 508
16'hD910 + #{8*index}	cmn_hns_pmu_mpam_pardid_mask0-7	RW	5.2.4.132 cmn_hns_pmu_mpam_pardid_mask0-7 on page 510
16'h3C00 : 16'h43E0	cmn_hns_rn_cluster0-63_physid_reg0	RW	5.2.4.133 cmn_hns_rn_cluster0-63_physid_reg0 on page 511
16'h4400 : 16'h4BE0	cmn_hns_rn_cluster64-127_physid_reg0	RW	5.2.4.134 cmn_hns_rn_cluster64-127_physid_reg0 on page 513
16'h3C08 : 16'h4BE8	cmn_hns_rn_cluster0-127_physid_reg1	RW	5.2.4.135 cmn_hns_rn_cluster0-127_physid_reg1 on page 514
16'h3C10 : 16'h4BF0	cmn_hns_rn_cluster0-127_physid_reg2	RW	5.2.4.136 cmn_hns_rn_cluster0-127_physid_reg2 on page 516
16'h3C18 : 16'h4BF8	cmn_hns_rn_cluster0-127_physid_reg3	RW	5.2.4.137 cmn_hns_rn_cluster0-127_physid_reg3 on page 518

Offset	Name	Type	Description
16'h5010 : 16'h51F8	cmn_hns_sam_nonhash_cfg1_memregion2-63	RW	5.2.4.138 cmn_hns_sam_nonhash_cfg1_memregion2-63 on page 520
16'h5210 : 16'h53F8	cmn_hns_sam_nonhash_cfg2_memregion2-63	RW	5.2.4.139 cmn_hns_sam_nonhash_cfg2_memregion2-63 on page 522
16'h5400 : 16'h5478	cmn_hns_sam_htg_cfg1_memregion0-15	RW	5.2.4.140 cmn_hns_sam_htg_cfg1_memregion0-15 on page 524
16'h5480 : 16'h54F8	cmn_hns_sam_htg_cfg2_memregion0-15	RW	5.2.4.141 cmn_hns_sam_htg_cfg2_memregion0-15 on page 525
16'h5500 : 16'h5578	cmn_hns_sam_htg_cfg3_memregion0-15	RW	5.2.4.142 cmn_hns_sam_htg_cfg3_memregion0-15 on page 526
16'h5600 : 16'h5678	cmn_hns_sam_htg_sn_nodeid_reg0-15	RW	5.2.4.143 cmn_hns_sam_htg_sn_nodeid_reg0-15 on page 528
16'h5680 : 16'h56F8	cmn_hns_sam_htg_sn_attr0-15	RW	5.2.4.144 cmn_hns_sam_htg_sn_attr0-15 on page 529
16'h5700 : 16'h5718	cmn_hns_sam_ccg_sa_nodeid_reg0-3	RW	5.2.4.145 cmn_hns_sam_ccg_sa_nodeid_reg0-3 on page 533
16'h5740 : 16'h5758	cmn_hns_sam_ccg_sa_attr0-3	RW	5.2.4.146 cmn_hns_sam_ccg_sa_attr0-3 on page 534
16'h5780 : 16'h57B8	hns_generic_regs0-7	RW	5.2.4.147 hns_generic_regs0-7 on page 538
16'h5900	cmn_hns_pa2setaddr_slc	RW	5.2.4.148 cmn_hns_pa2setaddr_slc on page 539
16'h5908	cmn_hns_pa2setaddr_sf	RW	5.2.4.149 cmn_hns_pa2setaddr_sf on page 542
16'h5910	cmn_hns_pa2setaddr_flex_slc	RW	5.2.4.150 cmn_hns_pa2setaddr_flex_slc on page 544
16'h5918	cmn_hns_pa2setaddr_flex_sf	RW	5.2.4.151 cmn_hns_pa2setaddr_flex_sf on page 546
{0-31} 16'h7000 : 16'h70F8	lcn_hashed_tgt_grp_cfg1_region0-31	RW	5.2.4.152 lcn_hashed_tgt_grp_cfg1_region0-31 on page 547
{0-31} 16'h7100 : 16'h71F8	lcn_hashed_tgt_grp_cfg2_region0-31	RW	5.2.4.153 lcn_hashed_tgt_grp_cfg2_region0-31 on page 548
{0-31} 16'h7200 : 16'h72F8	lcn_hashed_target_grp_secondary_cfg1_reg0-31	RW	5.2.4.154 lcn_hashed_target_grp_secondary_cfg1_reg0-31 on page 549
{0-31} 16'h7300 : 16'h73F8	lcn_hashed_target_grp_secondary_cfg2_reg0-31	RW	5.2.4.155 lcn_hashed_target_grp_secondary_cfg2_reg0-31 on page 551
{0-31} 16'h7400 : 16'h74F8	lcn_hashed_target_grp_hash_cntl_reg0-31	RW	5.2.4.156 lcn_hashed_target_grp_hash_cntl_reg0-31 on page 552
{0-3} 16'h7500 : 16'h7518	lcn_hashed_target_group_hn_count_reg0-3	RW	5.2.4.157 lcn_hashed_target_group_hn_count_reg0-3 on page 554
{0-7} 16'h7520 : 16'h7558	lcn_hashed_target_grp_cal_mode_reg0-7	RW	5.2.4.158 lcn_hashed_target_grp_cal_mode_reg0-7 on page 555
{0-1} 16'h7560 : 16'h7568	lcn_hashed_target_grp_hnf_cpa_en_reg0-1	RW	5.2.4.159 lcn_hashed_target_grp_hnf_cpa_en_reg0-1 on page 558
{0-15} 16'h7580 : 16'h75F8	lcn_hashed_target_grp_cpag_perhnf_reg0-15	RW	5.2.4.160 lcn_hashed_target_grp_cpag_perhnf_reg0-15 on page 559
{0-31} 16'h7700 : 16'h77F8	lcn_hashed_target_grp_compact_cpag_ctrl0-31	RW	5.2.4.161 lcn_hashed_target_grp_compact_cpag_ctrl0-31 on page 560

Offset	Name	Type	Description
{0-31} 16'h7800 : 16'h78F8	lcn_hashed_target_grp_compact_hash_ctrl0-31	RW	5.2.4.162 lcn_hashed_target_grp_compact_hash_ctrl0-31 on page 562

5.2 Register Description

The following contains descriptions on the registers used in CMN-700.

5.2.1 Configuration register descriptions

This section lists the configuration registers.

5.2.1.1 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

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-1: por_info_global

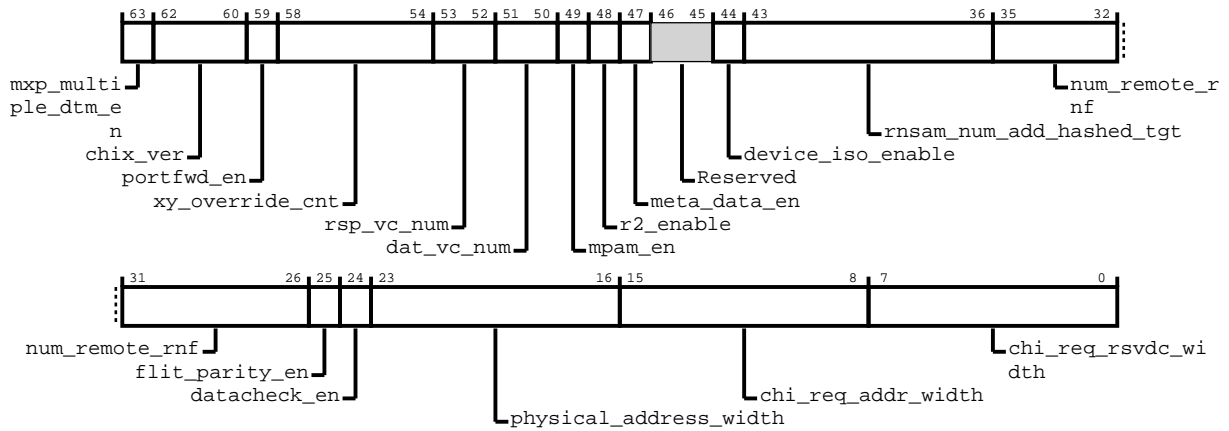


Table 5-5: por_info_global attributes

Bits	Name	Description	Type	Reset
[63]	mxp_multiple_dtm_en	Multiple DTMs feature enable. This is used if number of device ports on the XP is > 2	RO	Configuration dependent
[62:60]	chix_ver	CHIX Version Parameter: 2 -> CHIB, 3 -> CHIC, 4 -> CHID, 5 -> CHIE	RO	Configuration dependent
[59]	portfwd_en	CCIX Port to Port Forwarding feature enable	RO	Configuration dependent
[58:54]	xy_override_cnt	Number of Src-Tgt pairs whose XY route path can be overridden	RO	Configuration dependent
[53:52]	rsp_vc_num	Number of additional RSP channels internal to MXP. For increased bandwidth, this parameter need to be set to 2	RO	Configuration dependent
[51:50]	dat_vc_num	Number of additional DAT channels internal to MXP. For increased bandwidth, this parameter need to be set to 2	RO	Configuration dependent
[49]	mpam_en	MPAM enable	RO	Configuration dependent
[48]	r2_enable	CMN R2 feature enable	RO	Configuration dependent
[47]	meta_data_en	Meta Data Preservation mode enable	RO	Configuration dependent
[46:45]	Reserved	Reserved	RO	-
[44]	device_iso_enable	Disables smxp device ports	RO	Configuration dependent
[43:36]	rnsam_num_add_hashed_tgt	Number of additional hashed target ID's supported by the RN SAM, beyond the local HNF count	RO	Configuration dependent
[35:26]	num_remote_rnf	Number of remote RN-F devices in the system when the CML feature is enabled	RO	Configuration dependent
[25]	flit_parity_en	Indicates whether parity checking is enabled in the transport layer on all flits sent on the interconnect	RO	Configuration dependent
[24]	datacheck_en	Indicates whether datacheck feature is enabled for CHI DAT flit	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[23:16]	physical_address_width	Physical address 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

5.2.1.2 por_info_global_1

Contains user-specified values of build-time global configuration parameters.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h908

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-2: por_info_global_1

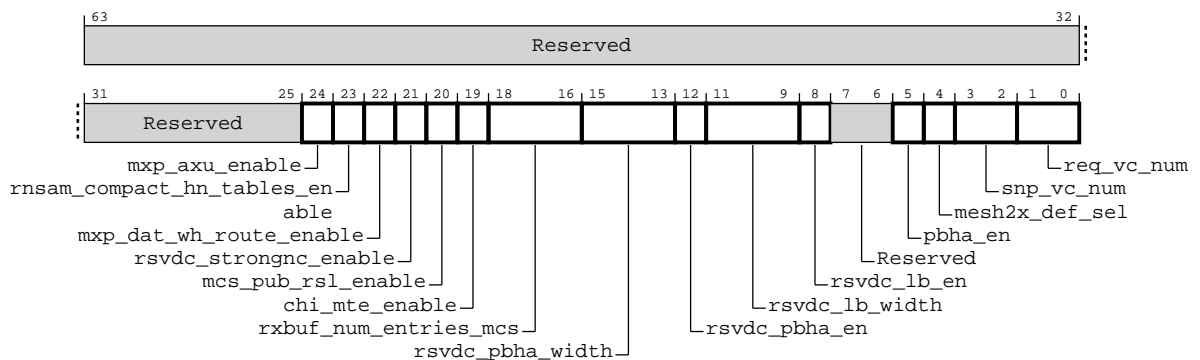


Table 5-6: por_info_global_1 attributes

Bits	Name	Description	Type	Reset
[63:25]	Reserved	Reserved	RO	-
[24]	mxp_axu_enable	MXP AXU interface Enable	RO	Configuration dependent
[23]	rnsam_compact_hn_tables_enable	RNSAM Compact HN Tables Enable	RO	Configuration dependent
[22]	mxp_dat_wh_route_enable	Worm Hole Routing Enable for MXP DAT channel	RO	Configuration dependent
[21]	rsvdc_strongnc_enable	RSVDC StrongNC Mode Enable	RO	Configuration dependent
[20]	mcs_pub_rsl_enable	Register Slice enable for MCS PUB outputs	RO	Configuration dependent
[19]	chi_mte_enable	CHI MTE Feature Enable	RO	Configuration dependent
[18:16]	rxbuf_num_entries_mcs	RX Buffer Entries at upload interface of MCSX/MCSY	RO	Configuration dependent
[15:13]	rsvdc_pbha_width	RSVDC PBHA Field Width	RO	Configuration dependent
[12]	rsvdc_pbha_en	RSVDC PBHA Mode Enable	RO	Configuration dependent
[11:9]	rsvdc_lb_width	RSVDC Loop Back Field Width	RO	Configuration dependent
[8]	rsvdc_lb_en	RSVDC Loop Back Mode Enable	RO	Configuration dependent
[7:6]	Reserved	Reserved	RO	-
[5]	pbha_en	PBHA Mode Enable	RO	Configuration dependent
[4]	mesh2x_def_sel	Default ping-pong scheme selection for TGTID Lookup in 2xMESH	RO	Configuration dependent
[3:2]	snp_vc_num	Number of additional SNP channels internal to MXP. For increased bandwidth, this parameter need to be set to 2	RO	Configuration dependent
[1:0]	req_vc_num	Number of additional REQ channels internal to MXP. For increased bandwidth, this parameter need to be set to 2	RO	Configuration dependent

5.2.1.3 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'h1900

Type

RW

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.ppu

Secure group override

por_cfgm_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the por_cfgm_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the por_cfgm_rcr.ppu bit and por_cfgm_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-3: por_ppu_int_enable

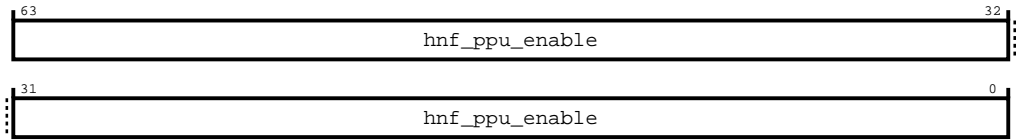


Table 5-7: por_ppu_int_enable attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_enable	Interrupt mask	RW	64'b0

5.2.1.4 por_ppu_int_enable_1

Configures the HN-F PPU event interrupt. Contains the interrupt mask.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1908

Type

RW

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.ppu

Secure group override

por_cfgm_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the por_cfgm_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the por_cfgm_rcr.ppu bit and por_cfgm_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-4: por_ppu_int_enable_1

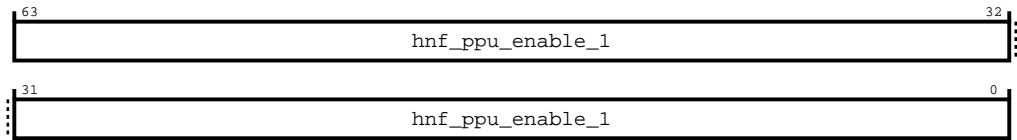


Table 5-8: por_ppu_int_enable_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_enable_1	Interrupt mask	RW	64'b0

5.2.1.5 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'h1910

Type

W1C

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.ppu

Secure group override

por_cfgm_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the por_cfgm_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the por_cfgm_rcr.ppu bit and por_cfgm_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-5: por_ppu_int_status

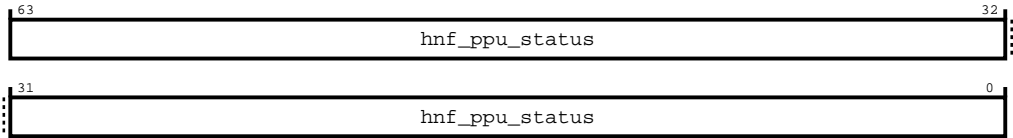


Table 5-9: por_ppu_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_status	Interrupt status	W1C	64'b0

5.2.1.6 por_ppu_int_status_1

Provides HN-F PPU event interrupt status.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1918

Type

W1C

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.ppu

Secure group override

por_cfgm_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the por_cfgm_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the por_cfgm_rcr.ppu bit and por_cfgm_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-6: por_ppu_int_status_1

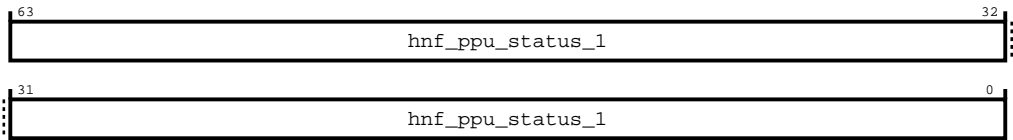


Table 5-10: por_ppu_int_status_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_ppu_status_1	Interrupt status	W1C	64'b0

5.2.1.7 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'h1920

Type

RW

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.ppu

Secure group override

por_cfgm_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the por_cfgm_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the por_cfgm_rcr.ppu bit and por_cfgm_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-7: por_ppu_qactive_hyst

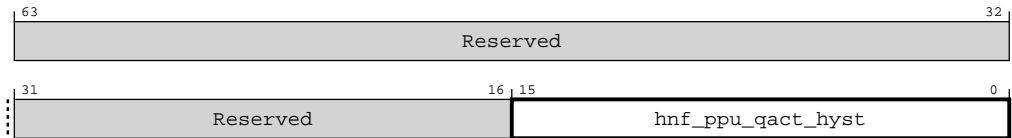


Table 5-11: por_ppu_qactive_hyst attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hnf_ppu_qact_hyst	QACTIVE hysteresis	RW	16'h10

5.2.1.8 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'h1928

Type

W1C

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.mpam

Secure group override

por_cfgm_scr.mpam

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions, unless the secure_override_default bit is set in the secure_control register: If the por_cfgm_scr.mpam bit is set, Non-secure and Realm accesses to this register are permitted

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-8: por_mpam_s_err_int_status

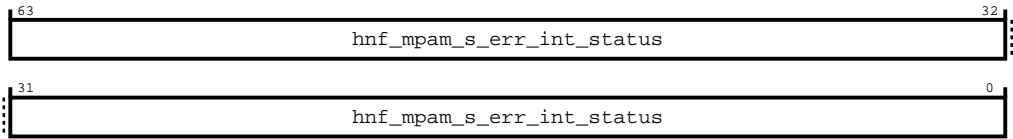


Table 5-12: por_mpam_s_err_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_s_err_int_status	MPAM S Interrupt status	W1C	64'b0

5.2.1.9 por_mpam_s_err_int_status_1

Provides HN-F MPAM Secure Error interrupt status.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1930

Type

W1C

Reset value

See individual bit resets

Root group override

por_cfgm_rcr.mpam

Secure group override

por_cfgm_scr.mpam

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions, unless the secure_override_default bit is set in the secure_control register:
If the por_cfgm_scr.mpam bit is set, Non-secure and Realm accesses to this register are permitted

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-9: por_mpam_s_err_int_status_1

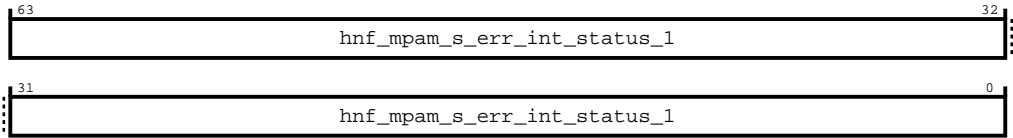


Table 5-13: por_mpam_s_err_int_status_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_s_err_int_status_1	MPAM S Interrupt status	W1C	64'b0

5.2.1.10 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'h1938

Type

W1C

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-10: por_mpam_ns_err_int_status

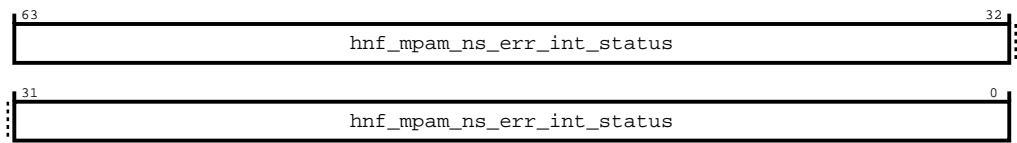


Table 5-14: por_mpam_ns_err_int_status attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_ns_err_int_status	MPAM NS Interrupt status	W1C	64'b0

5.2.1.11 por_mpam_ns_err_int_status_1

Provides HN-F MPAM Non-Secure Error interrupt status.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1940

Type

W1C

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-11: por_mpam_ns_err_int_status_1

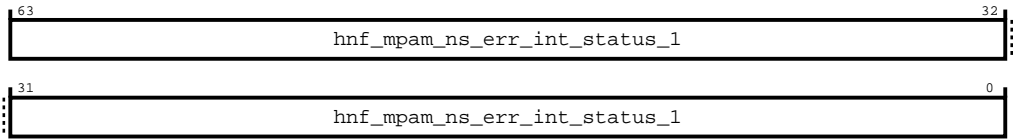


Table 5-15: por_mpam_ns_err_int_status_1 attributes

Bits	Name	Description	Type	Reset
[63:0]	hnf_mpam_ns_err_int_status_1	MPAM NS Interrupt status	W1C	64'b0

5.2.2 HN-S MPAM_S register descriptions

This section lists the HN-S MPAM_S registers.

5.2.2.1 cmn_hns_mpam_s_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-12: cmn_hns_mpam_s_node_info

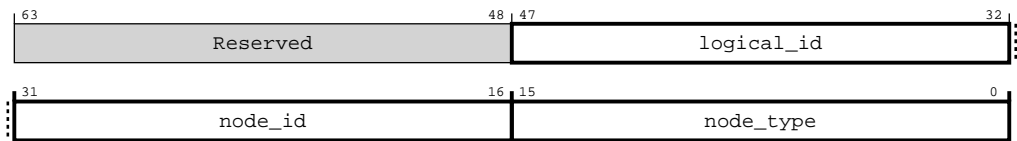


Table 5-16: cmn_hns_mpam_s_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	\$logical_id_description	RO	Configuration dependent
[31:16]	node_id	\$node_id_description	RO	Configuration dependent
[15:0]	node_type	\$node_type_description	RO	Configuration dependent

5.2.2.2 cmn_hns_mpam_s_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h80

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-13: cmn_hns_mpam_s_child_info

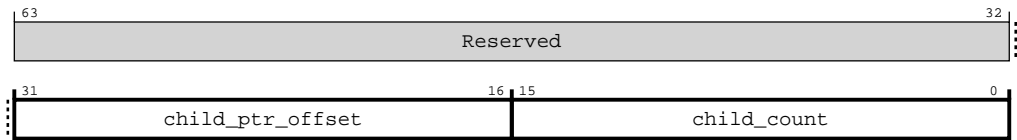


Table 5-17: cmn_hns_mpam_s_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

5.2.2.3 `cmn_hns_s_mpam_idr`

MPAM features ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1000

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-14: cmn_hns_s_mpam_idr

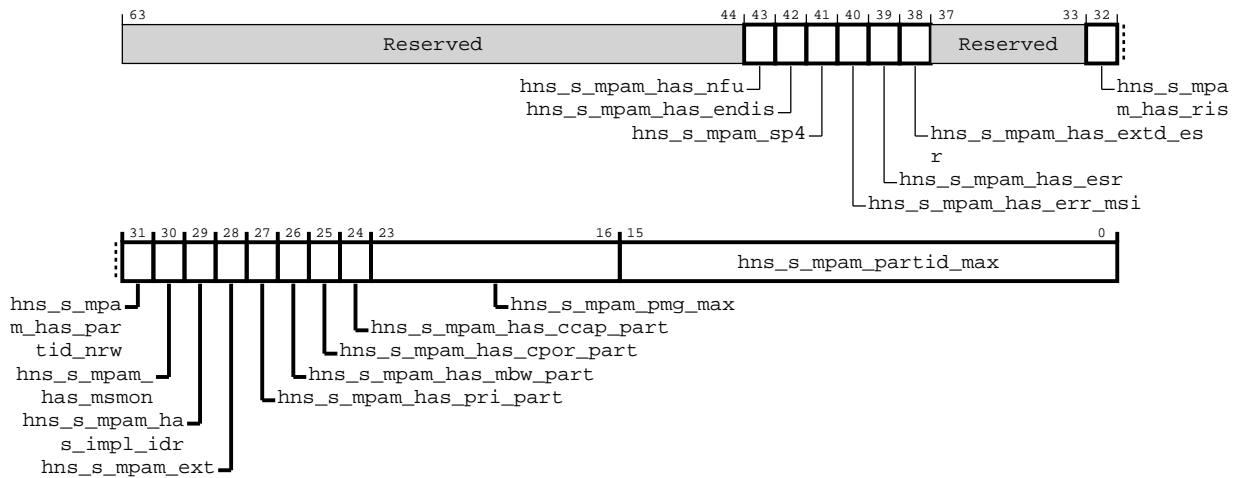


Table 5-18: cmn_hns_s_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_s_mpam_has_nfu	0: HN-F does not support no future use field 1: HN-F supports no future use field	RO	1'b0
[42]	hns_s_mpam_has_endis	0: HN-F does not support PARTID enable and disable functionality 1: HN-F supports PARTID enable and disable functionality	RO	1'b0
[41]	hns_s_mpam_sp4	0: HN-F supports two PARTID spaces 1: HN-F supports four PARTID spaces	RO	1'b1
[40]	hns_s_mpam_has_err_msi	0: HN-F does not support MSI writes to signal MPAM error interrupt 1: HN-F supports MSI writes to signal MPAM error interrupt	RO	1'b0
[39]	hns_s_mpam_has_esr	0: HN-F does not support MPAM error handling 1: HN-F supports MPAM error handling	RO	1'b1
[38]	hns_s_mpam_has_extd_esr	0: MPAMF_ESR is 32 bits 1: MPAMF_ESR is 64 bits	RO	1'b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_s_mpam_has_ris	0: HN-F does not support MPAM resource instance selector 1: HN-F supports MPAM resource instance selector	RO	1'b0
[31]	hns_s_mpam_has_partid_nrw	0: HN-F does not support MPAM PARTID Narrowing 1: HN-F supports MPAM PARTID Narrowing	RO	Configuration dependent
[30]	hns_s_mpam_has_msmon	0: MPAM performance monitoring is not supported 1: MPAM performance monitoring is supported	RO	Configuration dependent
[29]	hns_s_mpam_has_impl_idr	0: MPAM implementation specific partitioning features not supported 1: MPAM implementation specific partitioning features supported	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[28]	hns_s_mpam_ext	0: HN-F has no defined bits in [63:32] 1: HN-F has bits defined in [63:32]	RO	1'b1
[27]	hns_s_mpam_has_pri_part	0: MPAM priority partitioning is not supported 1: MPAM priority partitioning is supported	RO	Configuration dependent
[26]	hns_s_mpam_has_mbw_part	0: MPAM memory bandwidth partitioning is not supported 1: MPAM memory bandwidth partitioning is supported	RO	Configuration dependent
[25]	hns_s_mpam_has_cpor_part	0: MPAM cache portion partitioning is not supported 1: MPAM cache portion partitioning is supported	RO	Configuration dependent
[24]	hns_s_mpam_has_ccap_part	0: MPAM cache maximum capacity partitioning is not supported 1: MPAM cache maximum capacity partitioning is supported	RO	Configuration dependent
[23:16]	hns_s_mpam_pmg_max	Maximum value of non-secure PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_s_mpam_partid_max	Maximum value of secure PARTID supported by this HN-F	RO	Configuration dependent

5.2.2.4 cmn_hns_mpam_sidr

MPAM features Secure ID register. This is Secure (S) register only.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1008

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-15: cmn_hns_mpam_sidr

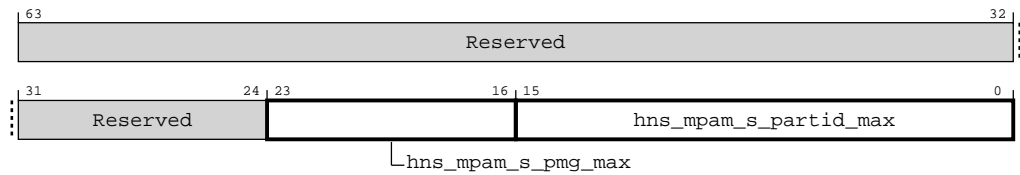


Table 5-19: cmn_hns_mpam_sidr attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_mpam_s_pmg_max	Maximum value of secure PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_mpam_s_partid_max	Maximum value of secure PARTID supported by this HN-F	RO	Configuration dependent

5.2.2.5 cmn_hns_s_mpam_iidr

MPAM Implementation ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1018

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-16: cmn_hns_s_mpam_iidr

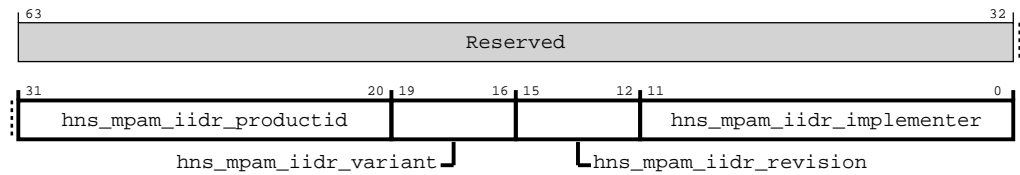


Table 5-20: cmn_hns_s_mpam_iidr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hns_mpam_iidr_productid	Implementation defined value identifying MPAM memory system component	RO	12'h000
[19:16]	hns_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	4'b0000
[15:12]	hns_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	4'b0000
[11:0]	hns_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	12'h43B

5.2.2.6 cmn_hns_s_mpam_iidr

MPAM architecture ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1020

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-17: cmn_hns_s_mpam_aidr

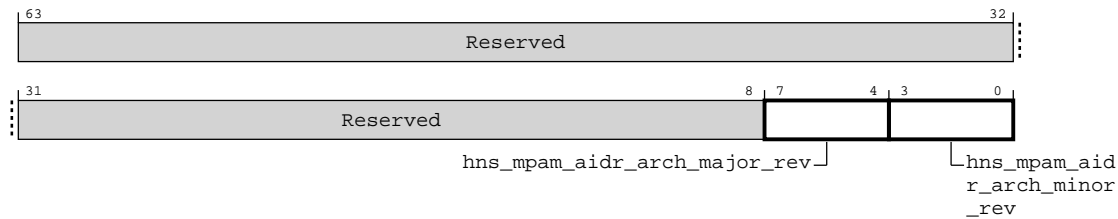


Table 5-21: cmn_hns_s_mpam_aidr attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hns_mpam_aidr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	4'b0001
[3:0]	hns_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	4'b0001

5.2.2.7 cmn_hns_s_mpam_impl_idr

MPAM Implementation defined partitioning feature ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1028

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-18: cmn_hns_s_mpam_impl_idr

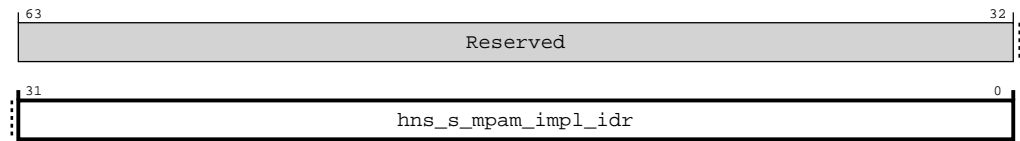


Table 5-22: cmn_hns_s_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hns_s_mpam_impl_idr	Implementation defined partitioning features.	RO	32'h00000000

5.2.2.8 cmn_hns_s_mpam_cpor_idr

MPAM cache portion partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1030

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-19: cmn_hns_s_mpam_cpor_idr

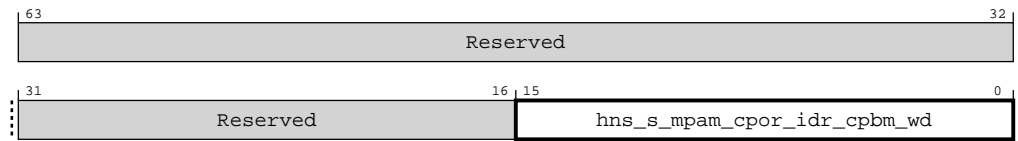


Table 5-23: cmn_hns_s_mpam_cpor_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bit map of this device.	RO	Configuration dependent

5.2.2.9 cmn_hns_s_mpam_ccap_idr

MPAM cache capacity partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1038

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-20: cmn_hns_s_mpam_ccap_idr

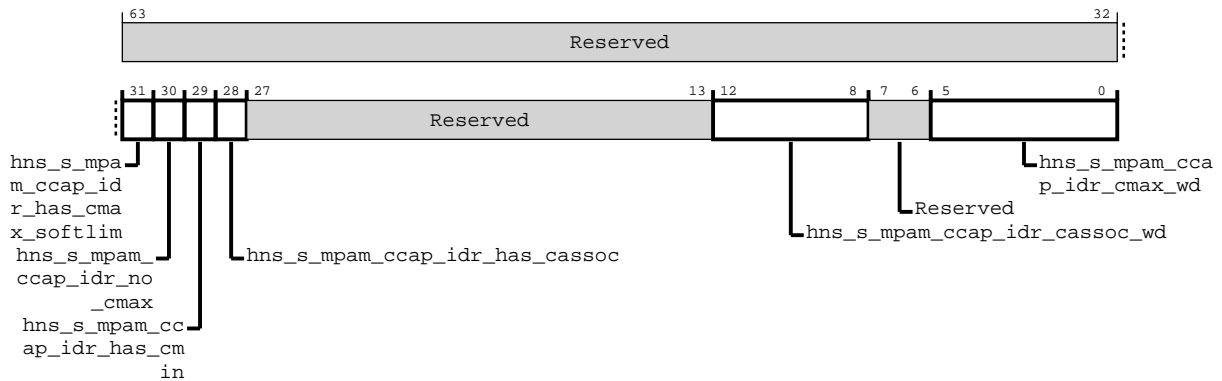


Table 5-24: cmn_hns_s_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_s_mpam_ccap_idr_has_cmax_softlim</code>	0: HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit 1: HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit	RO	1'h0
[30]	<code>hns_s_mpam_ccap_idr_no_cmax</code>	0: HN-F support MPAMCFG_CMAX 1: HN-F doesn't support MPAMCFG_CMAX	RO	1'h0
[29]	<code>hns_s_mpam_ccap_idr_has_cmin</code>	0: HN-F does not support MPAMCFG_CMIN 1: HN-F supports MPAMCFG_CMIN	RO	1'h0
[28]	<code>hns_s_mpam_ccap_idr_has_cassoc</code>	0: HN-F does not support MPAMCFG_CASSOC 1: HN-F supports MPAMCFG_CASSOC	RO	1'h0
[27:13]	Reserved	Reserved	RO	-
[12:8]	<code>hns_s_mpam_ccap_idr_cassoc_wd</code>	Number of fractional bits implemented in the cache associativity partitioning.	RO	5'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	<code>hns_s_mpam_ccap_idr_cmax_wd</code>	Number of fractional bits implemented in the cache capacity partitioning.	RO	Configuration dependent

5.2.2.10 cmn_hns_s_mpam_mbw_idr

MPAM Memory Bandwidth partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1040

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-21: cmn_hns_s_mpam_mbw_idr

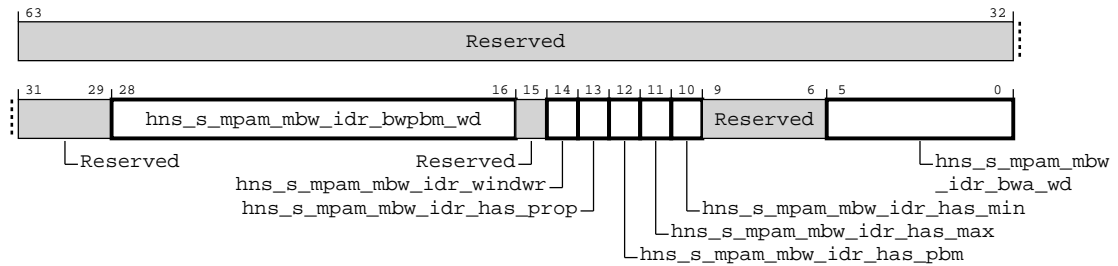


Table 5-25: cmn_hns_s_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hns_s_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register.	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hns_s_mpam_mbw_idr_windwr	0: The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed. 1: The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.	RO	1'h0
[13]	hns_s_mpam_mbw_idr_has_prop	0: There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register 1: MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.	RO	1'h0
[12]	hns_s_mpam_mbw_idr_has_pbm	0: There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register 1: MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.	RO	1'h0
[11]	hns_s_mpam_mbw_idr_has_max	0: There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register 1: MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.	RO	1'h0
[10]	hns_s_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

Bits	Name	Description	Type	Reset
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_s_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

5.2.2.11 cmn_hns_s_mpam_pri_idr

MPAM Priority partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1048

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-22: cmn_hns_s_mpam_pri_idr

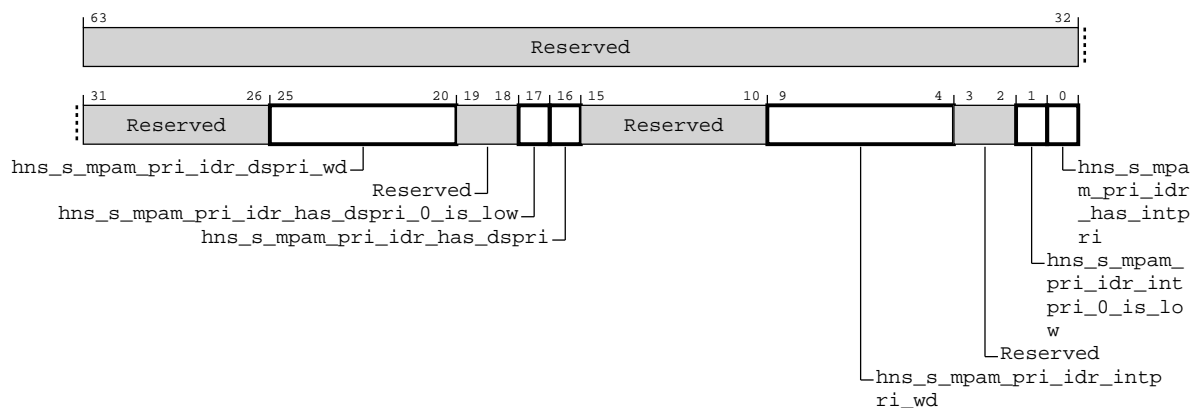


Table 5-26: cmn_hns_s_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	hns_s_mpam_pri_idr_dspri_wd	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	6'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_s_mpam_pri_idr_has_dspri_0_is_low	0: In the DSPRI field, a value of 0 means highest priority. 1: In the DSPRI field, a value of 0 means lowest priority.	RO	1'h0
[16]	hns_s_mpam_pri_idr_has_dspri	0: This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI. 1: This memory system component supports downstream priority and has an DSPRI field.	RO	1'h0
[15:10]	Reserved	Reserved	RO	-
[9:4]	hns_s_mpam_pri_idr_intpri_wd	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	6'h0
[3:2]	Reserved	Reserved	RO	-
[1]	hns_s_mpam_pri_idr_intpri_0_is_low	0: In the INTPRI field, a value of 0 means highest priority. 1: In the INTPRI field, a value of 0 means lowest priority.	RO	1'h0
[0]	hns_s_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

5.2.2.12 cmn_hns_s_mpam_partid_nrw_idr

MPAM PARTID narrowing ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1050

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-23: cmn_hns_s_mpam_partid_nrw_idr

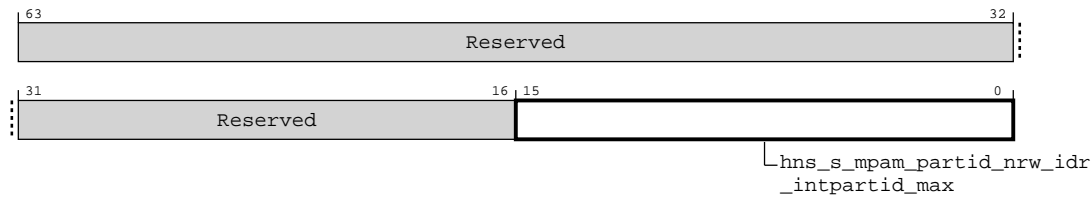


Table 5-27: cmn_hns_s_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

5.2.2.13 cmn_hns_s_mpam_msmon_idr

MPAM performance monitoring ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1080

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-24: cmn_hns_s_mpam_msmon_idr

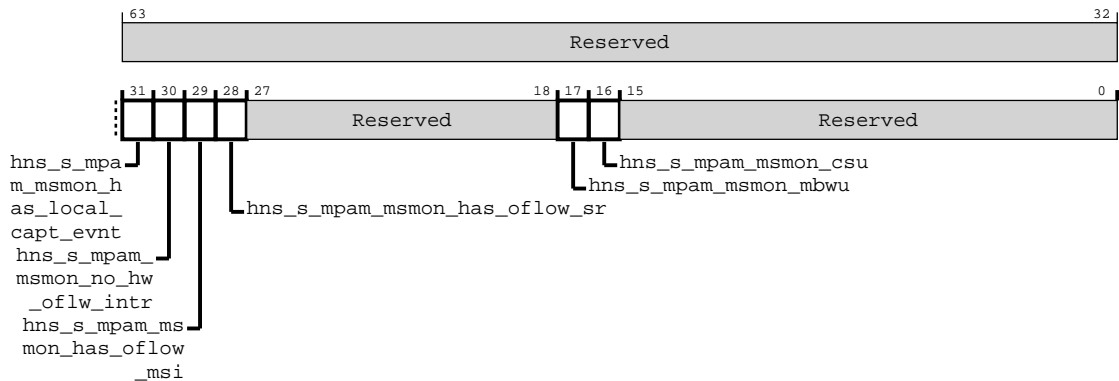


Table 5-28: cmn_hns_s_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_s_mpam_msmon_has_local_capt_evt</code>	Has the local capture event generator and the <code>MSMON_CAPT_EVT</code> register.	RO	1'h1
[30]	<code>hns_s_mpam_msmon_no_hw_oflw_intr</code>	0: HNF doesn't have hardwired MPAM overflow interrupt 1: HNF has have hardwired MPAM overflow interrupt	RO	1'b0
[29]	<code>hns_s_mpam_msmon_has_oflow_msi</code>	0: HNF doesn't have support for MSI writes to signal MPAM monitor overflow interrupt 1: HNF has support for MSI writes to signal MPAM monitor overflow interrupt	RO	1'b0
[28]	<code>hns_s_mpam_msmon_has_oflow_sr</code>	0: HNF doesn't have overflow status register 1: HNF has overflow status register	RO	1'b0
[27:18]	Reserved	Reserved	RO	-
[17]	<code>hns_s_mpam_msmon_mbwu</code>	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	<code>hns_s_mpam_msmon_csu</code>	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

5.2.2.14 cmn_hns_s_mpam_csumon_idr

MPAM cache storage usage monitor ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1088

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-25: cmn_hns_s_mpam_csumon_idr

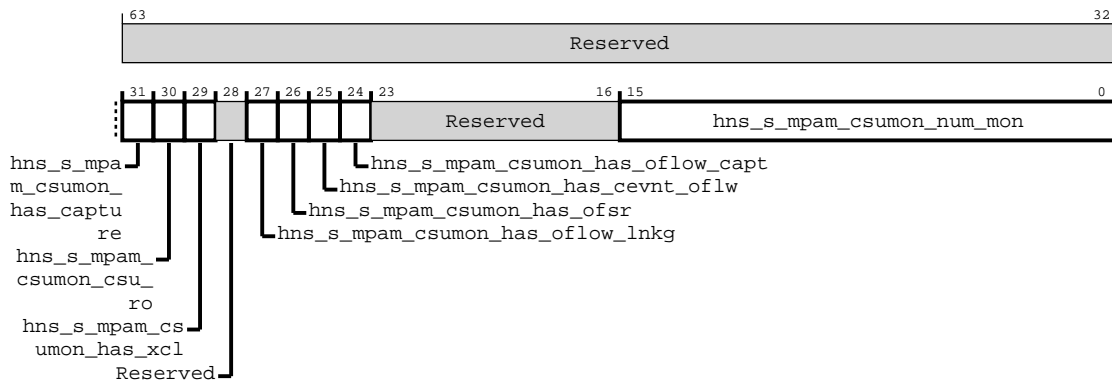


Table 5-29: cmn_hns_s_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_s_mpam_csumon_has_capture</code>	0: MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature. 1: This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.	RO	1'h1
[30]	<code>hns_s_mpam_csumon_csu_ro</code>	0: MSMON_CSU is read/write. 1: MSMON_CSU is read-only.	RO	1'b0
[29]	<code>hns_s_mpam_csumon_has_xcl</code>	0: MSMON_CFG_CSU_FLT does not implement the XCL field 1: MSMON_CFG_CSU_FLT implements the XCL field	RO	1'b0
[28]	Reserved	Reserved	RO	-
[27]	<code>hns_s_mpam_csumon_has_oflow_lnk</code>	0: HNF doesn't support CSU overflow linkage 1: HNF supports CSU overflow linkage	RO	1'b0
[26]	<code>hns_s_mpam_csumon_has_ofsr</code>	0: MSMON_CSU_OFSR register is not implemented 1: MSMON_CSU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_s_mpam_csumon_has_cevnt_oflw</code>	0: HNF doesn't support MSMON_CFG_CSU_CTL.CEVNT_OFLW 1: HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW	RO	1'b0

Bits	Name	Description	Type	Reset
[24]	hns_s_mpam_csumon_has_oflow_capt	0: HNF doesn't support MSMON_CFG_CSU_CTL.OFLOW_CAPT 1: HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT	RO	1'b0
[23:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	Configuration dependent

5.2.2.15 cmn_hns_s_mpam_mbwumon_idr

MPAM memory bandwidth usage monitor ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1090

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-26: cmn_hns_s_mpam_mbwumon_idr

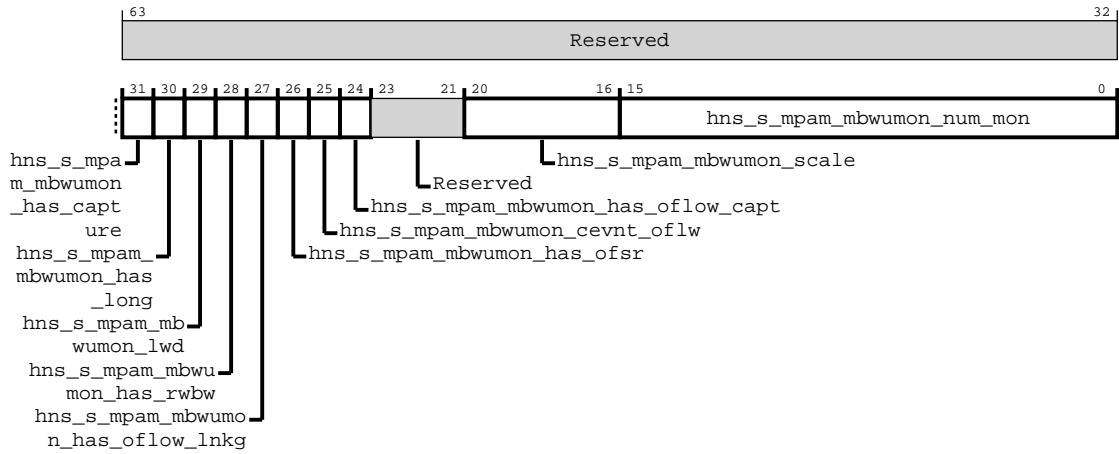


Table 5-30: cmn_hns_s_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_s_mpam_mbwumon_has_capture</code>	0: MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature. 1: This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.	RO	1'h0
[30]	<code>hns_s_mpam_mbwumon_has_long</code>	0: MSMON_MBWU_L is not implemented. 1: MSMON_MBWU_L is implemented.	RO	1'b0
[29]	<code>hns_s_mpam_mbwumon_lwd</code>	0: MSMON_MBWU_L has 44-bit VALUE field in bits [43:0]. 1: MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].	RO	1'b0
[28]	<code>hns_s_mpam_mbwumon_has_rwbw</code>	0: Read/write bandwidth selection is not implemented. 1: Read/write bandwidth selection is implemented.	RO	1'b0
[27]	<code>hns_s_mpam_mbwumon_has_oflow_lnkg</code>	0: Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_LNKG. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.	RO	1'b0
[26]	<code>hns_s_mpam_mbwumon_has_ofsr</code>	0: MSMON_MBWU_OFSR register is not implemented 1: MSMON_MBWU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_s_mpam_mbwumon_cevnt_oflw</code>	0: Doesn't support MSMON_CFG_MBWU_CTL.CEVNT_OFLW. 1: Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.	RO	1'b0
[24]	<code>hns_s_mpam_mbwumon_has_oflow_capt</code>	0: Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_CAPT. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.	RO	1'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	<code>hns_s_mpam_mbwumon_scale</code>	Scaling of MSMON_MBWU.VALUE in bits.	RO	5'h0
[15:0]	<code>hns_s_mpam_mbwumon_num_mon</code>	The number of MBWU monitoring counters implemented in this component.	RO	16'h0

5.2.2.16 cmn_hns_s_mpam_ecr

MPAM Error Control Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h10F0

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-27: cmn_hns_s_mpam_ecr

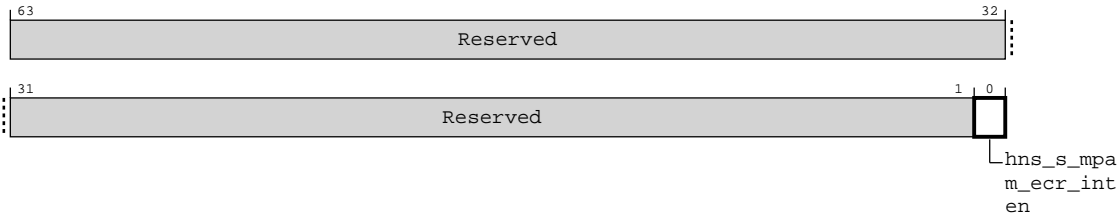


Table 5-31: cmn_hns_s_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hns_s_mpam_ecr_inten	Interrupt Enable. When INTEN = 0, MPAM error interrupts are not generated. When INTEN = 1, MPAM error interrupts are generated.	RW	1'h0

5.2.2.17 cmn_hns_s_mpam_esr

MPAM Error Status Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h10F8

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-28: cmn_hns_s_mpam_esr

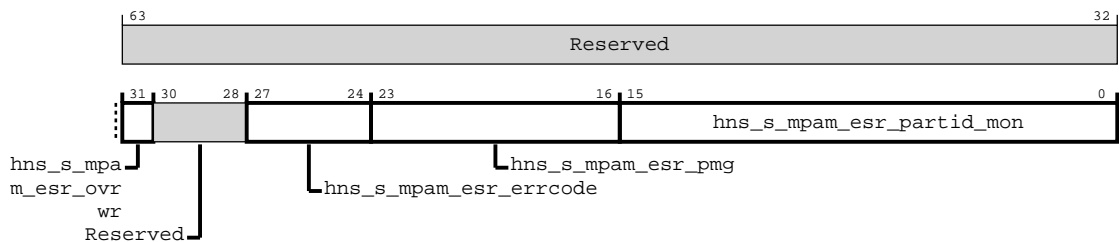


Table 5-32: cmn_hns_s_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_s_mpam_esr_ovrwr</code>	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	-

Bits	Name	Description	Type	Reset
[16]	hns_s_mpamcfg_part_sel_internal	If MPAMF_IDR.HAS_PARTID_NRW = 0, this field is RAZ/WI. If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interpret PARTID_SEL.	RW	1'h0
[15:0]	hns_s_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

5.2.2.21 cmn_hns_s_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1208

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-32: cmn_hns_s_mpamcfg_mbw_max

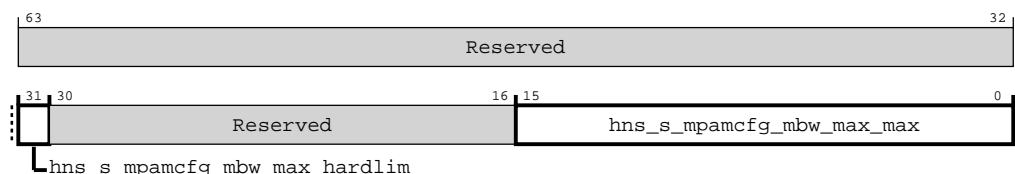


Table 5-36: cmn_hns_s_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_mpamcfg_mbw_max_hardlim	0: When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth. 1: When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_s_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.2.22 cmn_hns_s_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1220

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-33: cmn_hns_s_mpamcfg_mbw_winwd

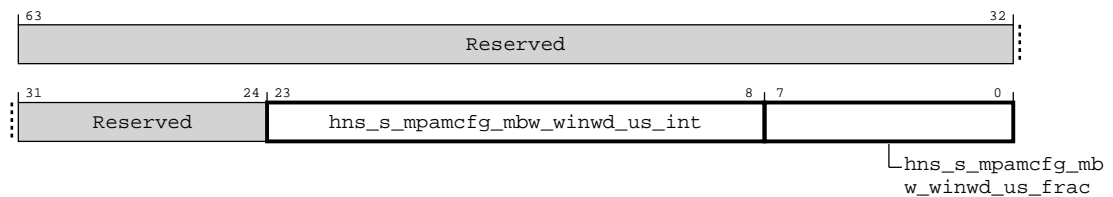


Table 5-37: cmn_hns_s_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hns_s_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hns_s_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

5.2.2.23 cmn_hns_s_mpamcfg_pri

MPAM priority partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1400

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-34: cmn_hns_s_mpamcfg_pri

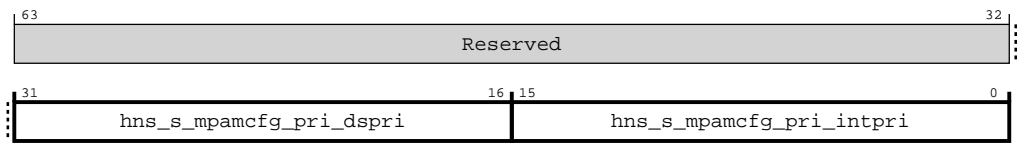


Table 5-38: cmn_hns_s_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	hns_s_mpamcfg_pri_dspri	If HAS_DSPRI is 1, this field is a priority value applied to downstream communications from this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	16'h0
[15:0]	hns_s_mpamcfg_pri_intpri	If HAS_INTPRI is 1, this field is a priority value applied internally inside this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.2.24 cmn_hns_s_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1500

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-35: cmn_hns_s_mpamcfg_mbw_prop

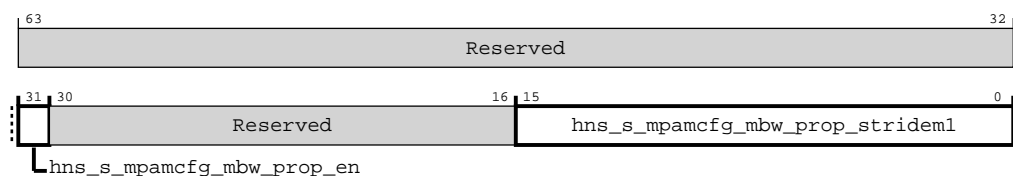


Table 5-39: cmn_hns_s_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_s_mpamcfg_mbw_prop_en</code>	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 <code>STRIDEM1</code> .	RW	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:0]	<code>hns_s_mpamcfg_mbw_prop_stridem1</code>	Normalized cost of a bandwidth consumption by the partition. <code>STRIDEM1</code> is the stride for the partition minus one.	RW	16'h0

5.2.2.25 cmn_hns_s_mpamcfg_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1600

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-36: cmn_hns_s_mpamcfg_intpartid

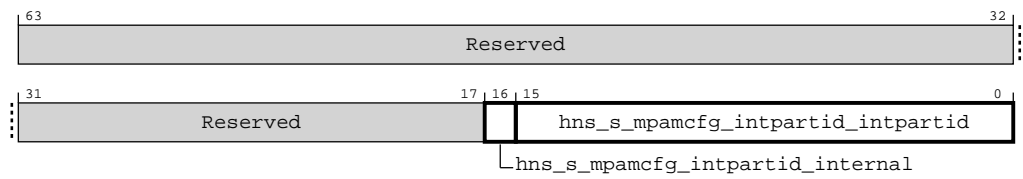


Table 5-40: cmn_hns_s_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_s_mpamcfg_intpartid_internal	This bit must be 1 when written to the register. If written as 0, the write will not update the reqPARTID to intPARTID association.	RW	1'h0
[15:0]	hns_s_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

5.2.2.26 cmn_hns_s_msmon_cfg_mon_sel

Memory system performance monitor selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1800

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-37: cmn_hns_s_msmon_cfg_mon_sel

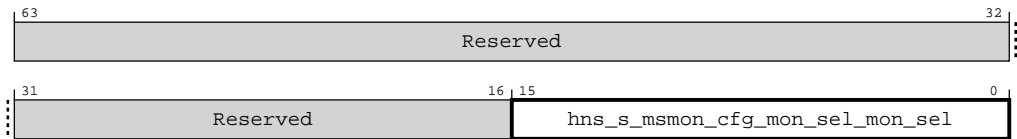


Table 5-41: cmn_hns_s_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure.	RW	16'h0

5.2.2.27 cmn_hns_s_msmon_capt_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1808

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-38: cmn_hns_s_msmon_capt_evnt

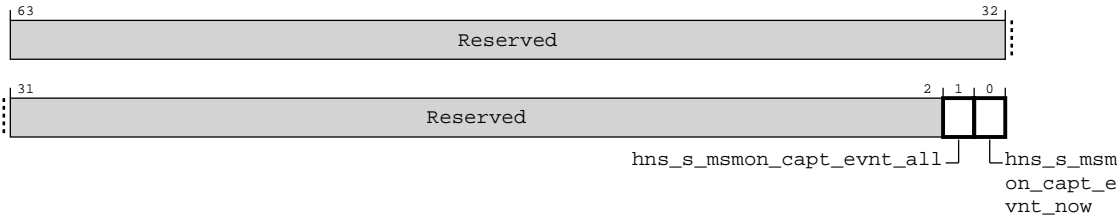


Table 5-42: cmn_hns_s_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1]	hns_s_msmon_capt_evnt_all	In secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to secure monitors in this memory system component with CAPT_EVNT = 7. In non-secure version if NOW is written as 1, signal a capture event to non-secure monitors in this memory system component with CAPT_EVNT = 7. In root version, if ALL written as 1 and NOW is also written as 1, signal a capture event to root, realm, secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to root monitors in this memory system component with CAPT_EVNT = 7. In realm version, if ALL written as 1 and NOW is also written as 1, signal a capture event to realm and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to realm monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hns_s_msmon_capt_evnt_now	When written as 1, this bit causes an event to all monitors in this memory system component with CAPT_EVNT set to the value of 7. When this bit is written as 0, no event is signalled.	RW	1'h0

5.2.2.28 cmn_hns_s_msmon_cfg_csuflt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1810

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-39: cmn_hns_s_msmon_cfg_csuflt

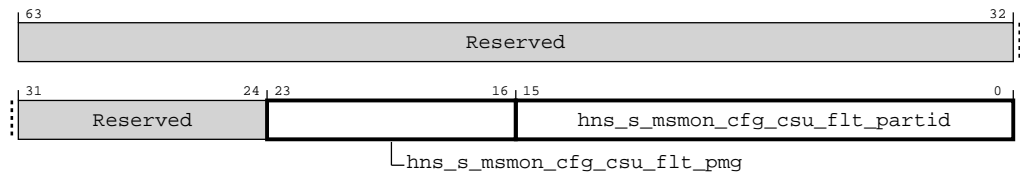


Table 5-43: cmn_hns_s_msmon_cfg_csuflt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_s_msmon_cfg_csuflt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_s_msmon_cfg_csuflt_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

5.2.2.29 cmn_hns_s_msmon_cfg_csuctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1818

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-40: cmn_hns_s_msmon_cfg_csu_ctl

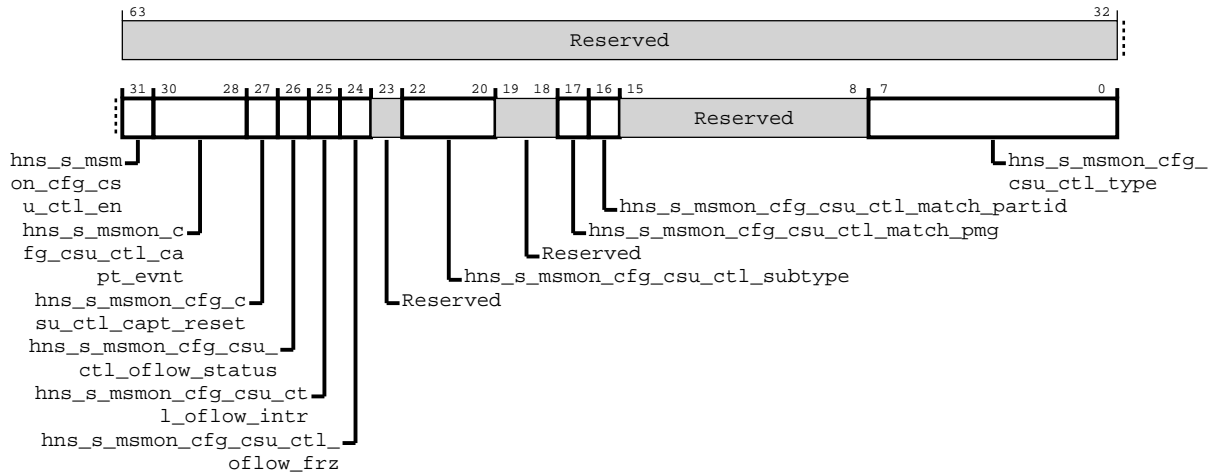


Table 5-44: cmn_hns_s_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_cfg_csu_ctl_en	0: The monitor is disabled and must not collect any information. 1: The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_s_msmon_cfg_csu_ctl_capt_evt	Select the event that triggers capture from the following: 0: No capture event is triggered. 1: External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_s_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	hns_s_msmon_cfg_csu_ctl_oflow_status	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_s_msmon_cfg_csu_ctl_oflow_intr	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_s_msmon_cfg_csu_ctl_oflow_frz	0: Monitor count wraps on overflow. 1: Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.	RW	1'h0
[23]	Reserved	Reserved	RO	-
[22:20]	hns_s_msmon_cfg_csu_ctl_subtype	Not currently used for CSU monitors, but reserved for future use.	RW	3'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_s_msmon_cfg_csu_ctl_match_pmg	0: Monitor storage used by all PMG values. 1: Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_s_msmon_cfg_csu_ctl_match_partid	0: Monitor storage used by all PARTIDs. 1: Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_s_msmon_cfg_csu_ctl_type	Read-only: Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	8'h43

5.2.2.30 cmn_hns_s_msmon_cfg_mbwuflt

Memory system performance monitor configure memory bandwidth usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1820

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-41: cmn_hns_s_msmon_cfg_mbwuflt

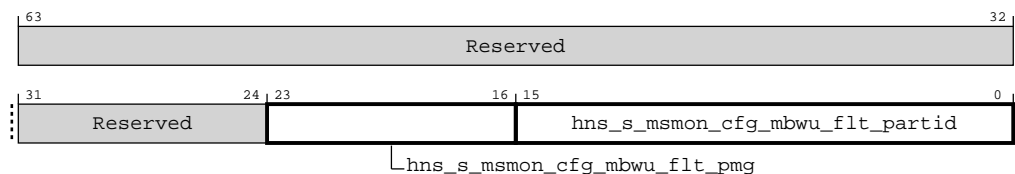


Table 5-45: cmn_hns_s_msmon_cfg_mbwuflt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_s_msmon_cfg_mbwuflt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_s_msmon_cfg_mbwuflt_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

5.2.2.31 cmn_hns_s_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.
This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1828

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-42: cmn_hns_s_msmon_cfg_mbwu_ctl

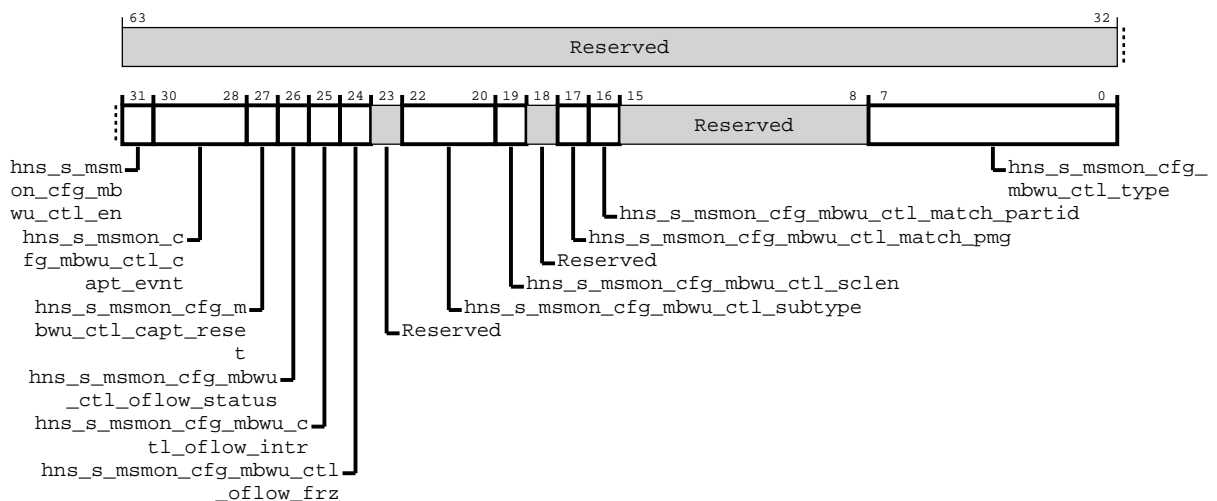


Table 5-46: cmn_hns_s_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_cfg_mbwu_ctl_en	0: The monitor is disabled and must not collect any information. 1: The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_s_msmon_cfg_mbwu_ctl_capt_evt	Select the event that triggers capture from the following: 0: No capture event is triggered. 1: External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_s_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hns_s_msmon_cfg_mbwu_ctl_oflow_status	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_s_msmon_cfg_mbwu_ctl_oflow_intr	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_s_msmon_cfg_mbwu_ctl_oflow_frz	0: Monitor count wraps on overflow. 1: Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.	RW	1'h0
[23]	Reserved	Reserved	RO	-
[22:20]	hns_s_msmon_cfg_mbwu_ctl_subtype	A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports: 0: Do not count any bandwidth. 1: Count bandwidth used by memory reads 2: Count bandwidth used by memory writes 3: Count bandwidth used by memory reads and memory writes All other values are reserved and behaviour of a monitor with SUBTYPE set to one of the reserved values is UNPREDICTABLE.	RW	3'h0
[19]	hns_s_msmon_cfg_mbwu_ctl_sclen	0: MSMON_MBWU.VALUE has bytes counted by the monitor instance. 1: MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.	RW	1'h0
[18]	Reserved	Reserved	RO	-
[17]	hns_s_msmon_cfg_mbwu_ctl_match_pmg	0: Monitor bandwidth used by all PMG values. 1: Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_s_msmon_cfg_mbwu_ctl_match_partid	0: Monitor bandwidth used by all PARTIDs. 1: Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_s_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

5.2.2.32 cmn_hns_s_msmon_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1840

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-43: cmn_hns_s_msmon_csu

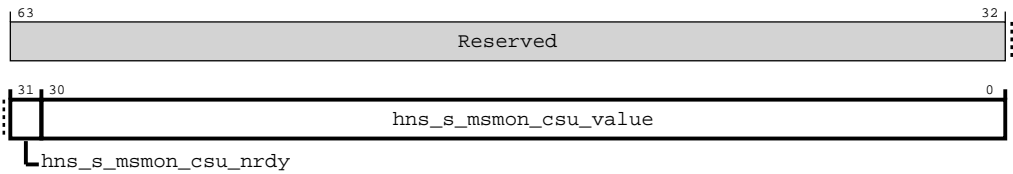


Table 5-47: cmn_hns_s_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_csu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_csu_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

5.2.2.33 cmn_hns_s_msmon_csu_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1848

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-44: cmn_hns_s_msmon_csu_capture

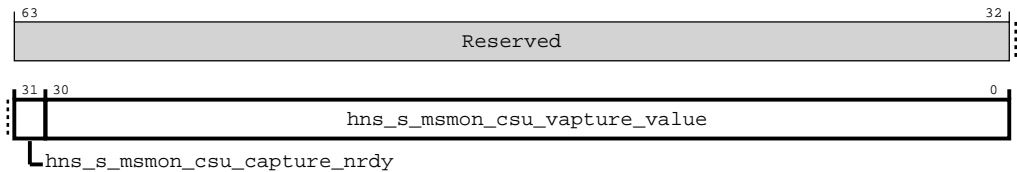


Table 5-48: cmn_hns_s_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_csu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_csu_vapture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

5.2.2.34 cmn_hns_s_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1860

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-45: cmn_hns_s_msmon_mbwu

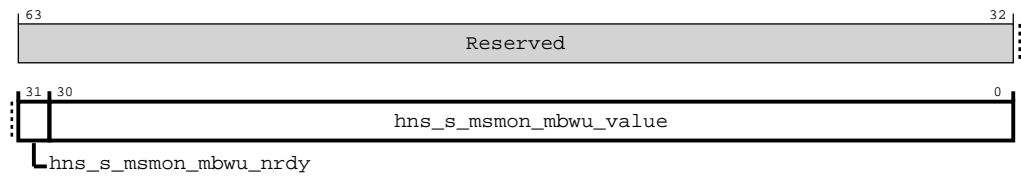


Table 5-49: cmn_hns_s_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_mbwu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_mbwu_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

5.2.2.35 cmn_hns_s_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1868

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-46: cmn_hns_s_msmon_mbwu_capture

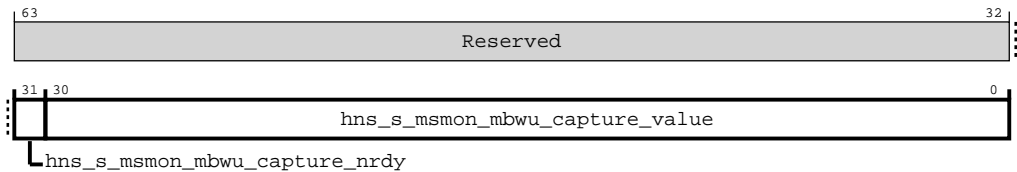


Table 5-50: cmn_hns_s_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_s_msmon_mbwu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_s_msmon_mbwu_capture_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

5.2.2.36 cmn_hns_s_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h2000

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-47: cmn_hns_s_mpamcfg_cpbm

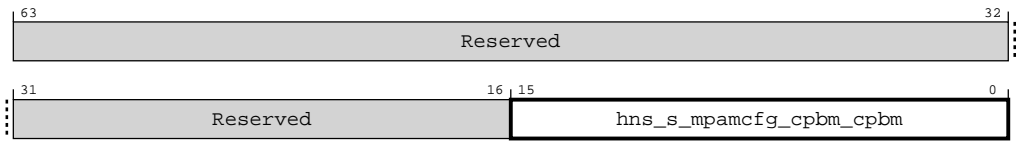


Table 5-51: cmn_hns_s_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_s_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. NOTE: CPBM can not be all zeros for any PARTID.	RW	16'hFFFF

5.2.2.37 cmn_hns_rt_mpam_idr

MPAM features ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8000

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-48: cmn_hns_rt_mpam_idr

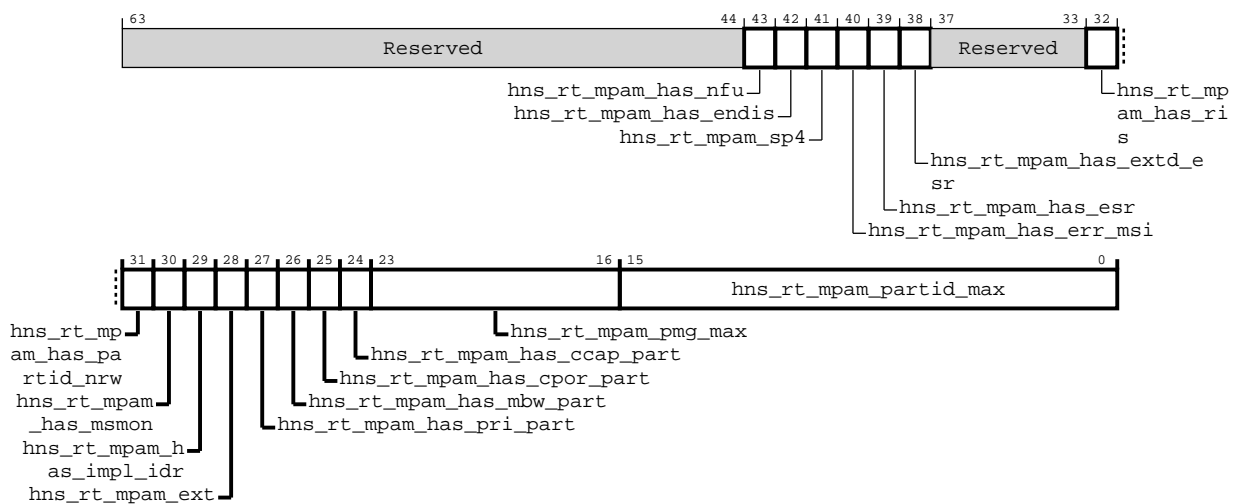


Table 5-52: cmn_hns_rt_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_rt_mpam_has_nfu	0: HN-F does not support no future use field 1: HN-F supports no future use field	RO	1'b0
[42]	hns_rt_mpam_has_endis	0: HN-F does not support PARTID enable and disable functionality 1: HN-F supports PARTID enable and disable functionality	RO	1'b0
[41]	hns_rt_mpam_sp4	0: HN-F supports two PARTID spaces 1: HN-F supports four PARTID spaces	RO	1'b1
[40]	hns_rt_mpam_has_err_msi	0: HN-F does not support MSI writes to signal MPAM error interrupt 1: HN-F supports MSI writes to signal MPAM error interrupt	RO	1'b0
[39]	hns_rt_mpam_has_esr	0: HN-F does not support MPAM error handling 1: HN-F supports MPAM error handling	RO	1'b1
[38]	hns_rt_mpam_has_extd_esr	0: MPAMF_ESR is 32 bits 1: MPAMF_ESR is 64 bits	RO	1'b0
[37:33]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[32]	hns_rt_mpam_has_ris	0: HN-F does not support MPAM resource instance selector 1: HN-F supports MPAM resource instance selector	RO	1'b0
[31]	hns_rt_mpam_has_partid_nrw	0: HN-F does not support MPAM PARTID Narrowing 1: HN-F supports MPAM PARTID Narrowing	RO	Configuration dependent
[30]	hns_rt_mpam_has_msmon	0: MPAM performance monitoring is not supported 1: MPAM performance monitoring is supported	RO	Configuration dependent
[29]	hns_rt_mpam_has_impl_idr	0: MPAM implementation specific partitioning features not supported 1: MPAM implementation specific partitioning features supported	RO	Configuration dependent
[28]	hns_rt_mpam_ext	0: HN-F has no defined bits in [63:32] 1: HN-F has bits defined in [63:32]	RO	1'b1
[27]	hns_rt_mpam_has_pri_part	0: MPAM priority partitioning is not supported 1: MPAM priority partitioning is supported	RO	Configuration dependent
[26]	hns_rt_mpam_has_mbw_part	0: MPAM memory bandwidth partitioning is not supported 1: MPAM memory bandwidth partitioning is supported	RO	Configuration dependent
[25]	hns_rt_mpam_has_cpor_part	0: MPAM cache portion partitioning is not supported 1: MPAM cache portion partitioning is supported	RO	Configuration dependent
[24]	hns_rt_mpam_has_ccap_part	0: MPAM cache maximum capacity partitioning is not supported 1: MPAM cache maximum capacity partitioning is supported	RO	Configuration dependent
[23:16]	hns_rt_mpam_pmg_max	Maximum value of root PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_rt_mpam_partid_max	Maximum value of root PARTID supported by this HN-F	RO	Configuration dependent

5.2.2.38 cmn_hns_rt_mpam_impl_idr

MPAM Implementation defined partitioning feature ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8028

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-49: cmn_hns_rt_mpam_impl_idr

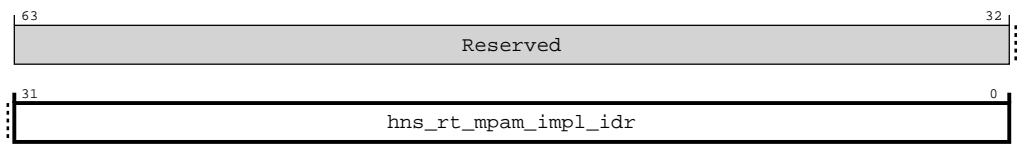


Table 5-53: cmn_hns_rt_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hns_rt_mpam_impl_idr	Implementation defined partitioning features.	RO	32'h00000000

5.2.2.39 cmn_hns_rt_mpam_cpor_idr

MPAM cache portion partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8030

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-50: cmn_hns_rt_mpam_cpor_idr

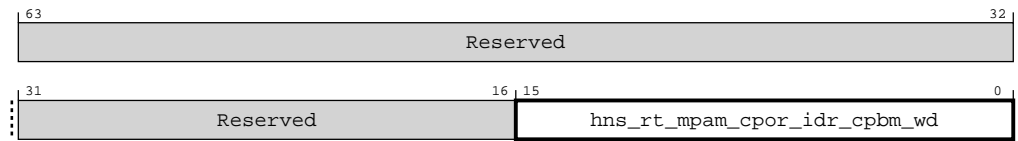


Table 5-54: cmn_hns_rt_mpam_cpor_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bit map of this device.	RO	Configuration dependent

5.2.2.40 cmn_hns_rt_mpam_ccap_idr

MPAM cache capacity partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8038

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-51: cmn_hns_rt_mpam_ccap_idr

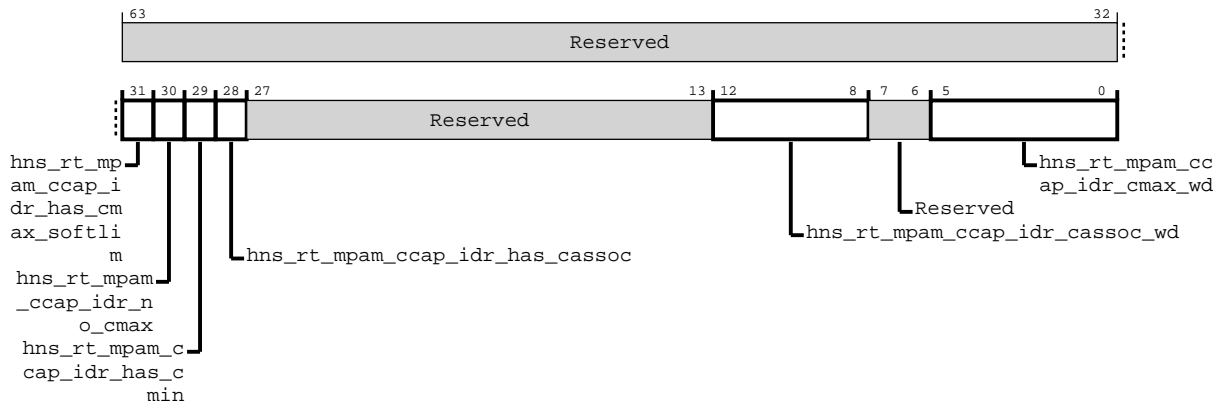


Table 5-55: cmn_hns_rt_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rt_mpam_ccap_idr_has_cmax_softlim</code>	0: HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit 1: HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit	RO	1'h0
[30]	<code>hns_rt_mpam_ccap_idr_no_cmax</code>	0: HN-F support MPAMCFG_CMAX 1: HN-F doesn't support MPAMCFG_CMAX	RO	1'h0
[29]	<code>hns_rt_mpam_ccap_idr_has_cmin</code>	0: HN-F does not support MPAMCFG_CMIN 1: HN-F supports MPAMCFG_CMIN	RO	1'h0
[28]	<code>hns_rt_mpam_ccap_idr_has_cassoc</code>	0: HN-F does not support MPAMCFG_CASSOC 1: HN-F supports MPAMCFG_CASSOC	RO	1'h0
[27:13]	Reserved	Reserved	RO	-
[12:8]	<code>hns_rt_mpam_ccap_idr_cassoc_wd</code>	Number of fractional bits implemented in the cache associativity partitioning.	RO	5'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	<code>hns_rt_mpam_ccap_idr_cmax_wd</code>	Number of fractional bits implemented in the cache capacity partitioning.	RO	Configuration dependent

5.2.2.41 cmn_hns_rt_mpam_mbw_idr

MPAM Memory Bandwidth partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8040

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-52: cmn_hns_rt_mpam_mbw_idr

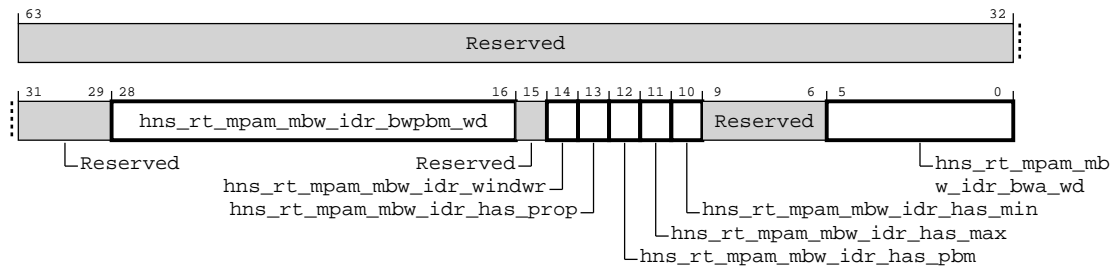


Table 5-56: cmn_hns_rt_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hns_rt_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register.	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hns_rt_mpam_mbw_idr_windwr	0: The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed. 1: The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.	RO	1'h0
[13]	hns_rt_mpam_mbw_idr_has_prop	0: There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register 1: MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.	RO	1'h0
[12]	hns_rt_mpam_mbw_idr_has_pbm	0: There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register 1: MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.	RO	1'h0
[11]	hns_rt_mpam_mbw_idr_has_max	0: There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register 1: MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.	RO	1'h0
[10]	hns_rt_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	-

Bits	Name	Description	Type	Reset
[5:0]	hns_rt_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

5.2.2.42 cmn_hns_rt_mpam_pri_idr

MPAM Priority partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8048

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-53: cmn_hns_rt_mpam_pri_idr

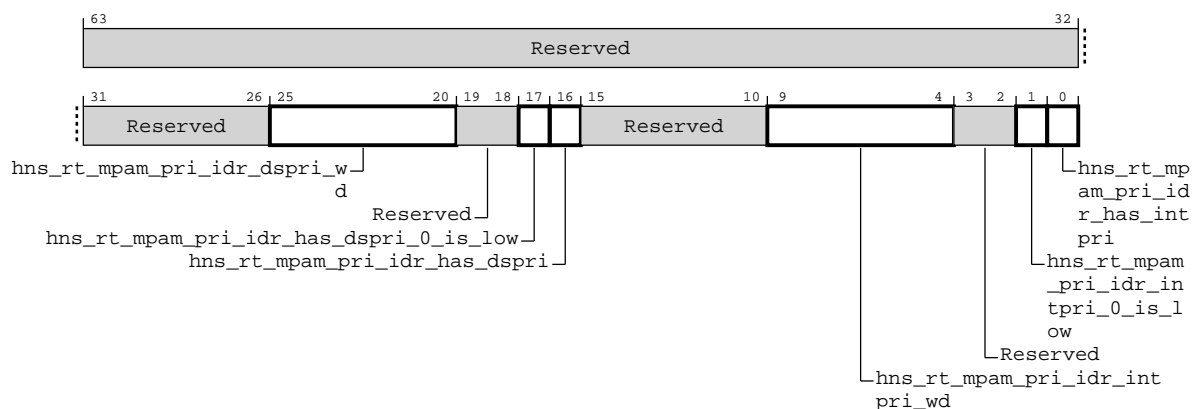


Table 5-57: cmn_hns_rt_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	hns_rt_mpam_pri_idr_dspri_wd	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	6'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_rt_mpam_pri_idr_has_dspri_0_is_low	0: In the DSPRI field, a value of 0 means highest priority. 1: In the DSPRI field, a value of 0 means lowest priority.	RO	1'h0
[16]	hns_rt_mpam_pri_idr_has_dspri	0: This memory system component supports priority, but doesn't have a downstream priority (DSPRI) field in MPAMCFG_PRI. 1: This memory system component supports downstream priority and has an DSPRI field.	RO	1'h0
[15:10]	Reserved	Reserved	RO	-
[9:4]	hns_rt_mpam_pri_idr_intpri_wd	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	6'h0
[3:2]	Reserved	Reserved	RO	-
[1]	hns_rt_mpam_pri_idr_intpri_0_is_low	0: In the INTPRI field, a value of 0 means highest priority. 1: In the INTPRI field, a value of 0 means lowest priority.	RO	1'h0
[0]	hns_rt_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

5.2.2.43 cmn_hns_rt_mpam_partid_nrw_idr

MPAM PARTID narrowing ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8050

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-54: cmn_hns_rt_mpam_partid_nrw_idr

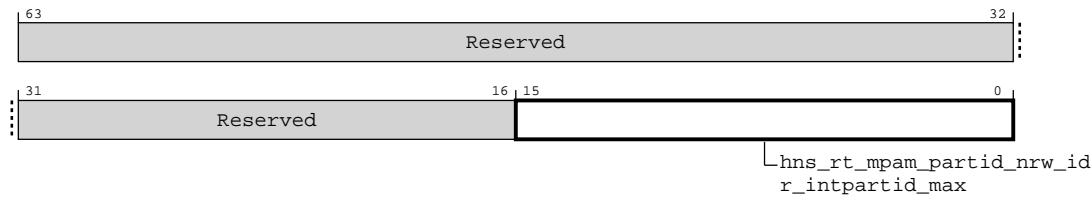


Table 5-58: cmn_hns_rt_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

5.2.2.44 cmn_hns_rt_mpam_msmon_idr

MPAM performance monitoring ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8080

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-55: cmn_hns_rt_mpam_msmon_idr

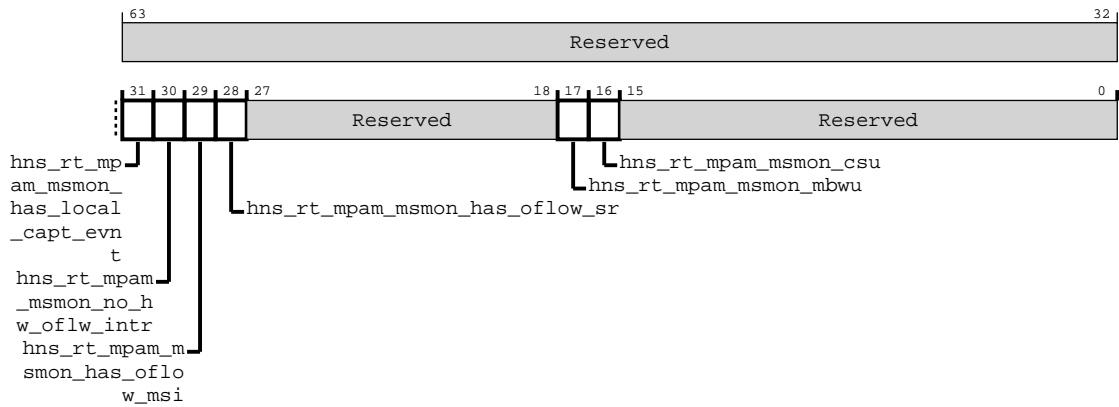


Table 5-59: cmn_hns_rt_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rt_mpam_msmon_has_local_capt_evt</code>	Has the local capture event generator and the MSMON_CAPT_EVT register.	RO	1'h1
[30]	<code>hns_rt_mpam_msmon_no_hw_oflw_intr</code>	0: HNF doesn't have hardwired MPAM overflow interrupt 1: HNF has have hardwired MPAM overflow interrupt	RO	1'b0
[29]	<code>hns_rt_mpam_msmon_has_oflow_msi</code>	0: HNF doesn't have support for MSI writes to signal MPAM monitor overflow interrupt 1: HNF has support for MSI writes to signal MPAM monitor overflow interrupt	RO	1'b0
[28]	<code>hns_rt_mpam_msmon_has_oflow_sr</code>	0: HNF doesn't have overflow status register 1: HNF has overflow status register	RO	1'b0
[27:18]	Reserved	Reserved	RO	-
[17]	<code>hns_rt_mpam_msmon_mbwu</code>	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	<code>hns_rt_mpam_msmon_csu</code>	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

5.2.2.45 cmn_hns_rt_mpam_csumon_idr

MPAM cache storage usage monitor ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8088

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-56: cmn_hns_rt_mpam_csumon_idr

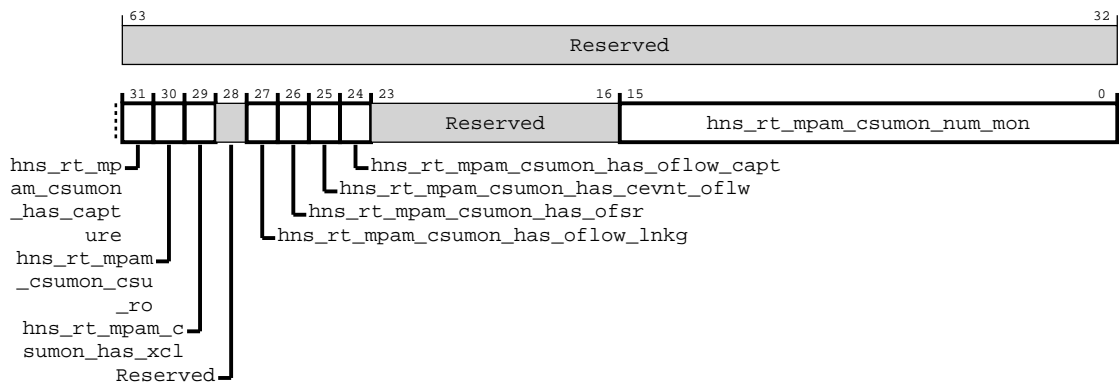


Table 5-60: cmn_hns_rt_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rt_mpam_csumon_has_capture</code>	0: MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature. 1: This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.	RO	1'h1
[30]	<code>hns_rt_mpam_csumon_csu_ro</code>	0: MSMON_CSU is read/write. 1: MSMON_CSU is read-only.	RO	1'b0
[29]	<code>hns_rt_mpam_csumon_has_xcl</code>	0: MSMON_CFG_CSU_FLT does not implement the XCL field 1: MSMON_CFG_CSU_FLT implements the XCL field	RO	1'b0
[28]	Reserved	Reserved	RO	-
[27]	<code>hns_rt_mpam_csumon_has_oflow_lnk</code>	0: HNF doesn't support CSU overflow linkage 1: HNF supports CSU overflow linkage	RO	1'b0
[26]	<code>hns_rt_mpam_csumon_has_ofsr</code>	0: MSMON_CSU_OFSR register is not implemented 1: MSMON_CSU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_rt_mpam_csumon_has_cevnt_oflw</code>	0: HNF doesn't support MSMON_CFG_CSU_CTL.CEVNT_OFLW 1: HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW	RO	1'b0

Bits	Name	Description	Type	Reset
[24]	hns_rt_mpam_csumon_has_oflow_capt	0: HNF doesn't support MSMON_CFG_CSU_CTL.OFLOW_CAPT 1: HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT	RO	1'b0
[23:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	Configuration dependent

5.2.2.46 cmn_hns_rt_mpam_mbwumon_idr

MPAM memory bandwidth usage monitor ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8090

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-57: cmn_hns_rt_mpam_mbwumon_idr

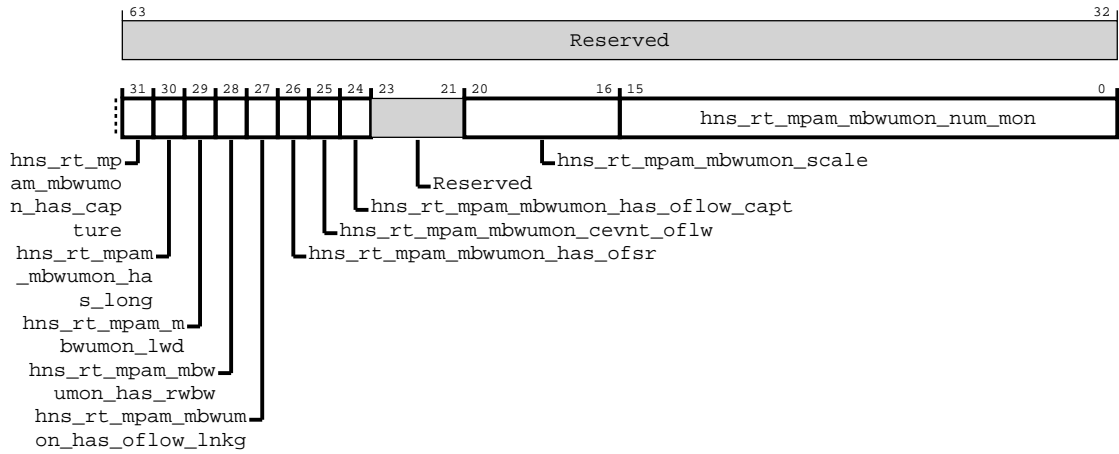


Table 5-61: cmn_hns_rt_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rt_mpam_mbwumon_has_capture</code>	0: MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature. 1: This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behaviour.	RO	1'h0
[30]	<code>hns_rt_mpam_mbwumon_has_long</code>	0: MSMON_MBWU_L is not implemented. 1: MSMON_MBWU_L is implemented.	RO	1'b0
[29]	<code>hns_rt_mpam_mbwumon_lwd</code>	0: MSMON_MBWU_L has 44-bit VALUE field in bits [43:0]. 1: MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].	RO	1'b0
[28]	<code>hns_rt_mpam_mbwumon_has_rwbw</code>	0: Read/write bandwidth selection is not implemented. 1: Read/write bandwidth selection is implemented.	RO	1'b0
[27]	<code>hns_rt_mpam_mbwumon_has_oflow_lnkg</code>	0: Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_LNKG. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.	RO	1'b0
[26]	<code>hns_rt_mpam_mbwumon_has_ofsr</code>	0: MSMON_MBWU_OFSR register is not implemented 1: MSMON_MBWU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_rt_mpam_mbwumon_cevnt_oflw</code>	0: Doesn't support MSMON_CFG_MBWU_CTL.CEVNT_OFLW. 1: Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.	RO	1'b0
[24]	<code>hns_rt_mpam_mbwumon_has_oflow_capt</code>	0: Doesn't support MSMON_CFG_MBWU_CTL.OFLOW_CAPT. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.	RO	1'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	<code>hns_rt_mpam_mbwumon_scale</code>	Scaling of MSMON_MBWU.VALUE in bits.	RO	5'h0
[15:0]	<code>hns_rt_mpam_mbwumon_num_mon</code>	The number of MBWU monitoring counters implemented in this component.	RO	16'h0

5.2.2.47 cmn_hns_rt_mpam_ecr

MPAM Error Control Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h80F0

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-58: cmn_hns_rt_mpam_ecr

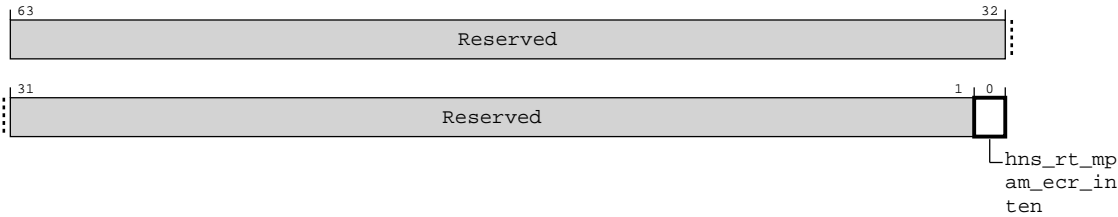


Table 5-62: cmn_hns_rt_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hns_rt_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

5.2.2.48 cmn_hns_rt_mpam_esr

MPAM Error Status Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h80F8

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-59: cmn_hns_rt_mpam_esr

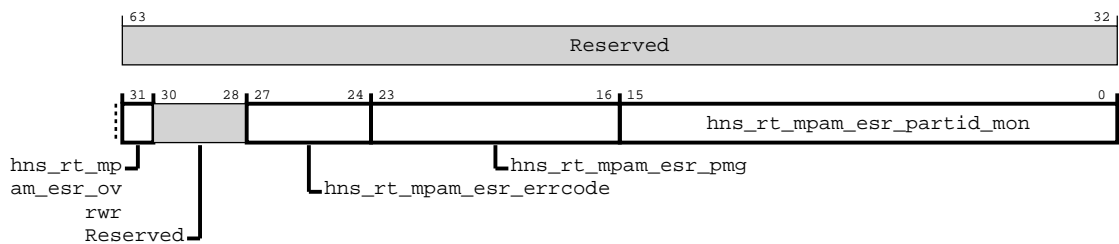


Table 5-63: cmn_hns_rt_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_mpam_esr_ovrwr	Overwritten. If 0 and ERRCODE is zero, no errors have occurred. If 0 and ERRCODE is non-zero, a single error has occurred and is recorded in this register. If 1 and ERRCODE is non-zero, multiple errors have occurred and this register records the most recent error. The state where this bit is 1 and ERRCODE is zero is not produced by hardware and is only reached when software writes this combination into this register.	RW	1'h0
[30:28]	Reserved	Reserved	RO	-
[27:24]	hns_rt_mpam_esr_errcode	Error code	RW	4'h0

Bits	Name	Description	Type	Reset
[23:16]	hns_rt_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hns_rt_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

5.2.2.49 cmn_hns_rt_mpamcfg_part_sel

MPAM partition configuration selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8100

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-60: cmn_hns_rt_mpamcfg_part_sel

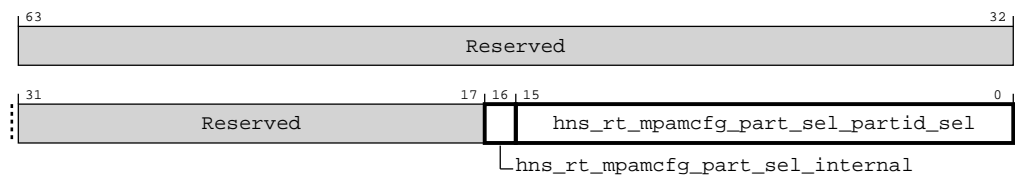


Table 5-64: cmn_hns_rt_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_rt_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

Bits	Name	Description	Type	Reset
[15:0]	hns_rt_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

5.2.2.50 cmn_hns_rt_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8108

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-61: cmn_hns_rt_mpamcfg_cmax

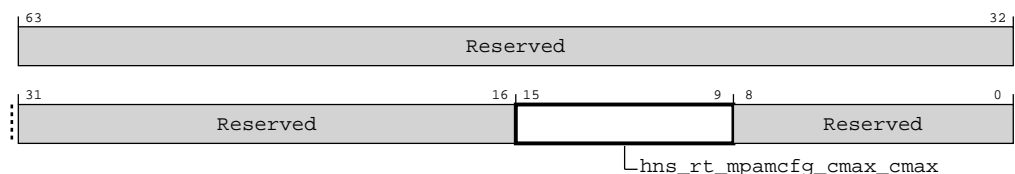


Table 5-65: cmn_hns_rt_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:9]	hns_rt_mpamcfg_cmax_cmax	Maximum cache capacity usage in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL.	RW	7'b1111111
[8:0]	Reserved	Reserved	RO	-

5.2.2.51 cmn_hns_rt_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8200

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-62: cmn_hns_rt_mpamcfg_mbw_min

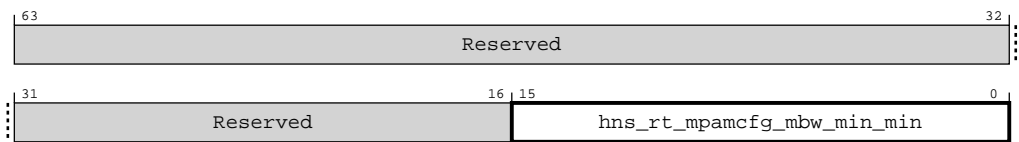


Table 5-66: cmn_hns_rt_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpamcfg_mbw_min_min	Memory minimum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.2.52 cmn_hns_rt_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8208

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-63: cmn_hns_rt_mpamcfg_mbw_max

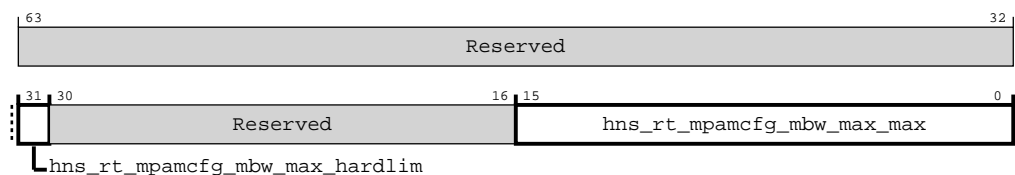


Table 5-67: cmn_hns_rt_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rt_mpamcfg_mbw_max_hardlim</code>	0: When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth. 1: When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:0]	<code>hns_rt_mpamcfg_mbw_max_max</code>	Memory maximum bandwidth allocated to the partition selected by <code>MPAMCFG_PART_SEL</code> .	RW	16'h0

5.2.2.53 cmn_hns_rt_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8220

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-64: cmn_hns_rt_mpamcfg_mbw_winwd

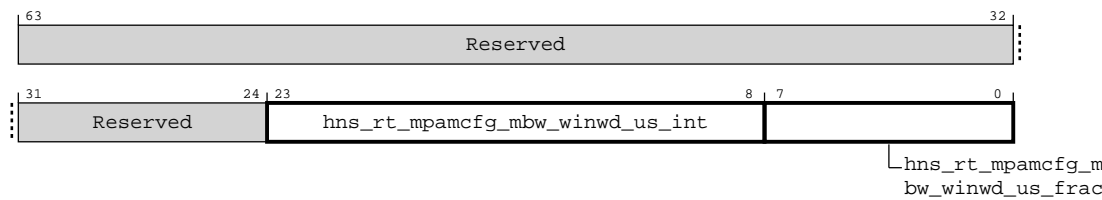


Table 5-68: cmn_hns_rt_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hns_rt_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hns_rt_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

5.2.2.54 cmn_hns_rt_mpamcfg_pri

MPAM priority partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8400

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-65: cmn_hns_rt_mpamcfg_pri

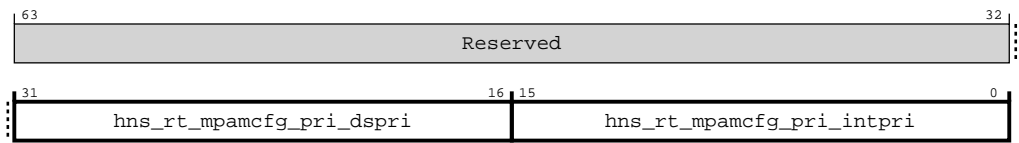


Table 5-69: cmn_hns_rt_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	hns_rt_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	16'h0
[15:0]	hns_rt_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	16'h0

5.2.2.55 cmn_hns_rt_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8500

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-66: cmn_hns_rt_mpamcfg_mbw_prop

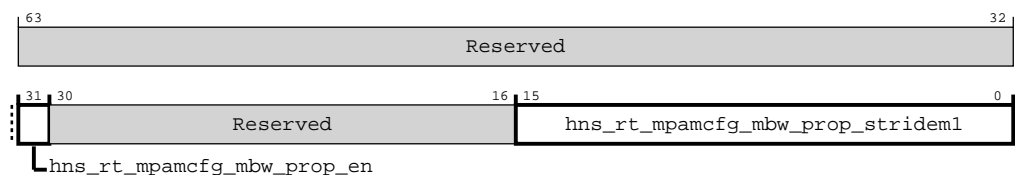


Table 5-70: cmn_hns_rt_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_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
[30:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpamcfg_mbw_prop_stridem1	Normalized cost of a bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	16'h0

5.2.2.56 cmn_hns_rt_mpamcfg_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8600

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-67: cmn_hns_rt_mpamcfg_intpartid

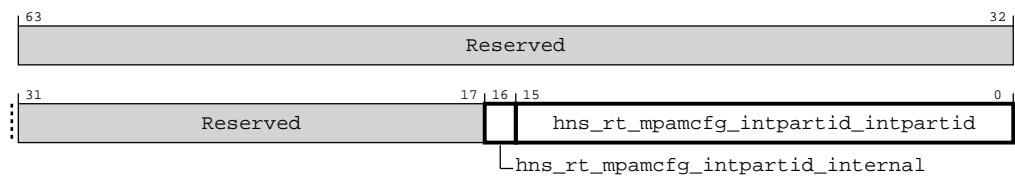


Table 5-71: cmn_hns_rt_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	hns_rt_mpamcfg_intpartid_internal	This bit must be 1 when written to the register. If written as 0, the write will not update the reqPARTID to intPARTID association.	RW	1'h0
[15:0]	hns_rt_mpamcfg_intpartid_intpartid	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

5.2.2.57 cmn_hns_rt_msmon_cfg_mon_sel

Memory system performance monitor selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8800

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-68: cmn_hns_rt_msmon_cfg_mon_sel

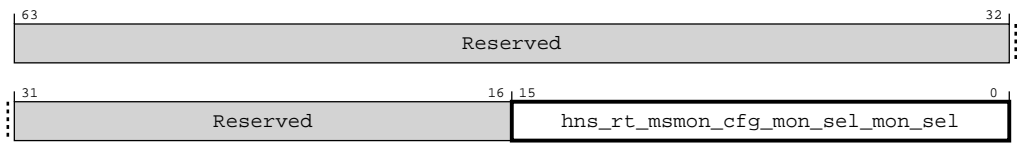


Table 5-72: cmn_hns_rt_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure.	RW	16'h0

5.2.2.58 cmn_hns_rt_msmon_capt_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8808

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-69: cmn_hns_rt_msmon_capt_evnt

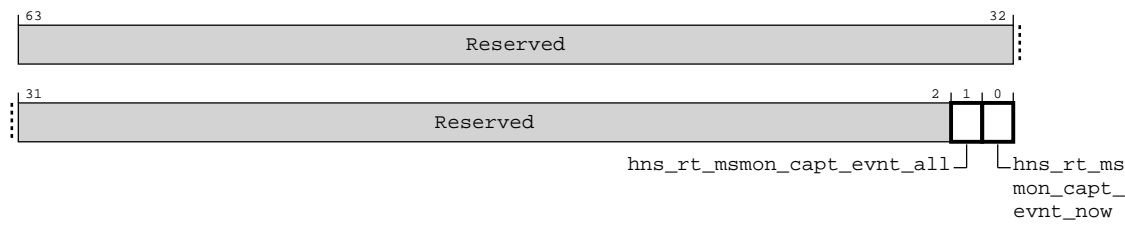


Table 5-73: cmn_hns_rt_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1]	hns_rt_msmon_capt_evnt_all	In secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to secure monitors in this memory system component with CAPT_EVNT = 7. In non-secure version if NOW is written as 1, signal a capture event to non-secure monitors in this memory system component with CAPT_EVNT = 7. In root version, if ALL written as 1 and NOW is also written as 1, signal a capture event to root, realm, secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to root monitors in this memory system component with CAPT_EVNT = 7. In realm version, if ALL written as 1 and NOW is also written as 1, signal a capture event to realm and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to realm monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hns_rt_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

5.2.2.59 cmn_hns_rt_msmon_cfg_csuflt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8810

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-70: cmn_hns_rt_msmon_cfg_csu_flt

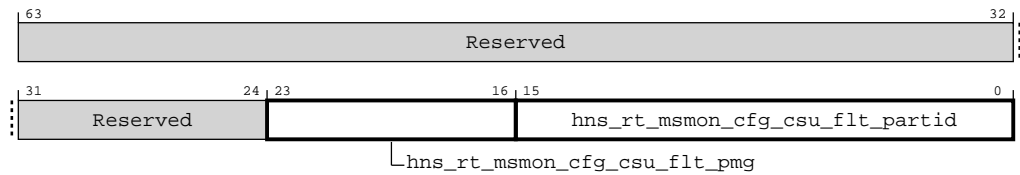


Table 5-74: cmn_hns_rt_msmon_cfg_csu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_rt_msmon_cfg_csu_flt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_rt_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

5.2.2.60 cmn_hns_rt_msmon_cfg_csu_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8818

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-71: cmn_hns_rt_msmon_cfg_csu_ctl

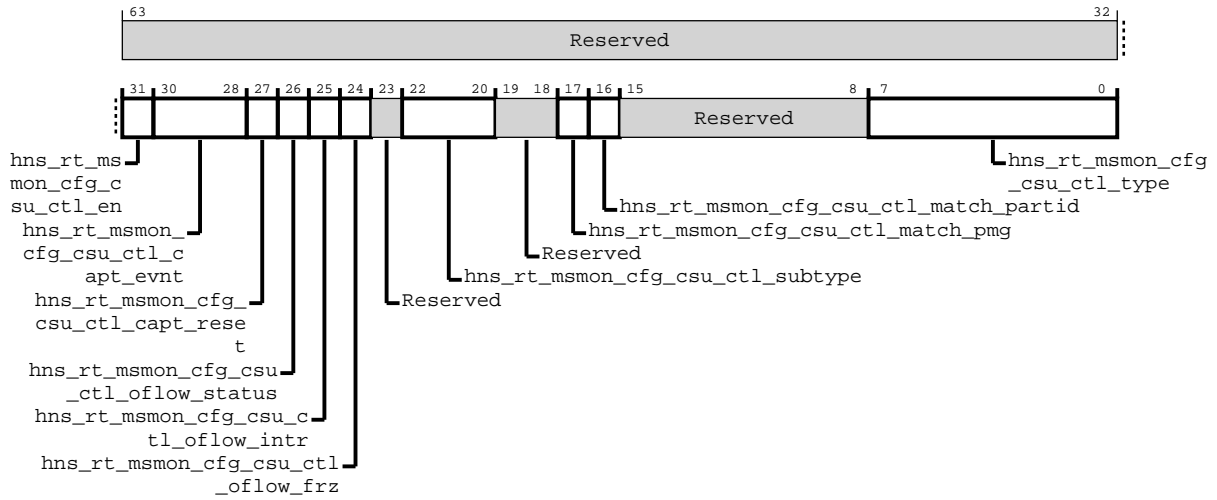


Table 5-75: cmn_hns_rt_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_msmon_cfg_csu_ctl_en	0: The monitor is disabled and must not collect any information. 1: The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_rt_msmon_cfg_csu_ctl_capt_evnt	Select the event that triggers capture from the following: 0: No capture event is triggered. 1: External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_rt_msmon_cfg_csu_ctl_capt_reset	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	hns_rt_msmon_cfg_csu_ctl_oflow_status	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_rt_msmon_cfg_csu_ctl_oflow_intr	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_rt_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]	Reserved	Reserved	RO	-
[22:20]	hns_rt_msmon_cfg_csu_ctl_subtype	Not currently used for CSU monitors, but reserved for future use.	RW	3'h0
[19:18]	Reserved	Reserved	RO	-
[17]	hns_rt_msmon_cfg_csu_ctl_match_pmg	0: Monitor storage used by all PMG values. 1: Only monitor storage used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_rt_msmon_cfg_csu_ctl_match_partid	0: Monitor storage used by all PARTIDs. 1: Only monitor storage used with the PARTID matching MSMON_CFG_CSU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_rt_msmon_cfg_csu_ctl_type	Read-only: Constant type indicating the type of the monitor. CSU monitor is TYPE = 0x43.	RW	8'h43

5.2.2.61 cmn_hns_rt_msmon_cfg_mbwuflt

Memory system performance monitor configure memory bandwidth usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8820

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-72: cmn_hns_rt_msmon_cfg_mbwuflt

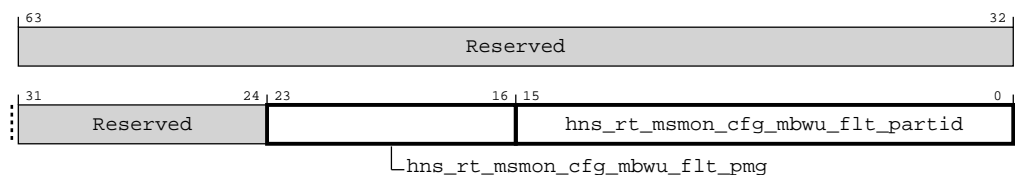


Table 5-76: cmn_hns_rt_msmon_cfg_mbwuflt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_rt_msmon_cfg_mbwuflt_pmg	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_rt_msmon_cfg_mbwuflt_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

5.2.2.62 cmn_hns_rt_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.
This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8828

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-73: cmn_hns_rt_msmon_cfg_mbwu_ctl

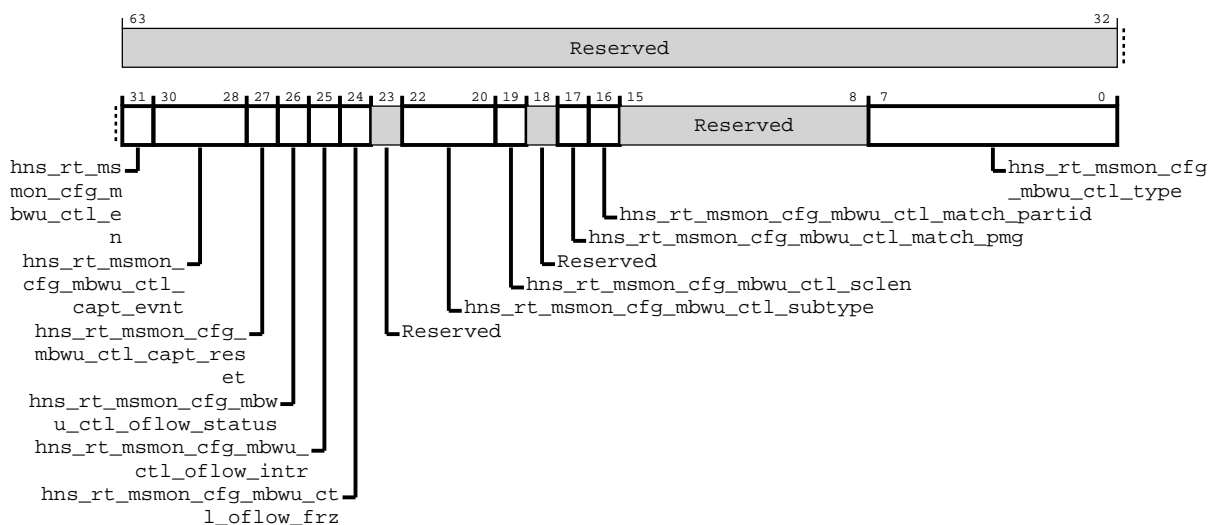


Table 5-77: cmn_hns_rt_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_msmon_cfg_mbwu_ctl_en	0: The monitor is disabled and must not collect any information. 1: The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_rt_msmon_cfg_mbwu_ctl_capt_evt	Select the event that triggers capture from the following: 0: No capture event is triggered. 1: External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_rt_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hns_rt_msmon_cfg_mbwu_ctl_oflow_status	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_rt_msmon_cfg_mbwu_ctl_oflow_intr	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_rt_msmon_cfg_mbwu_ctl_oflow_frz	0: Monitor count wraps on overflow. 1: Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.	RW	1'h0
[23]	Reserved	Reserved	RO	-
[22:20]	hns_rt_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	3'h0
[19]	hns_rt_msmon_cfg_mbwu_ctl_sclen	0: MSMON_MBWU.VALUE has bytes counted by the monitor instance. 1: MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.	RW	1'h0
[18]	Reserved	Reserved	RO	-
[17]	hns_rt_msmon_cfg_mbwu_ctl_match_pmg	0: Monitor bandwidth used by all PMG values. 1: Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_rt_msmon_cfg_mbwu_ctl_match_partid	0: Monitor bandwidth used by all PARTIDs. 1: Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_rt_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

5.2.2.63 cmn_hns_rt_msmon_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8840

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-74: cmn_hns_rt_msmon_csu

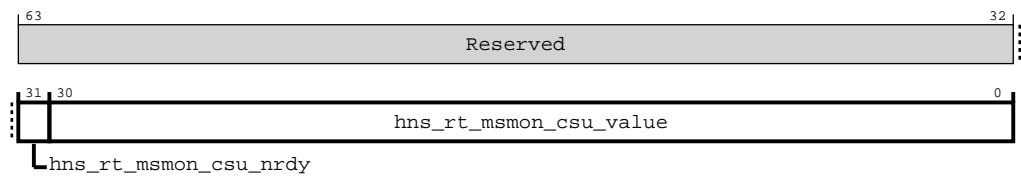


Table 5-78: cmn_hns_rt_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_msmon_csu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rt_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

5.2.2.64 cmn_hns_rt_msmon_csu_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8848

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-75: cmn_hns_rt_msmon_csu_capture

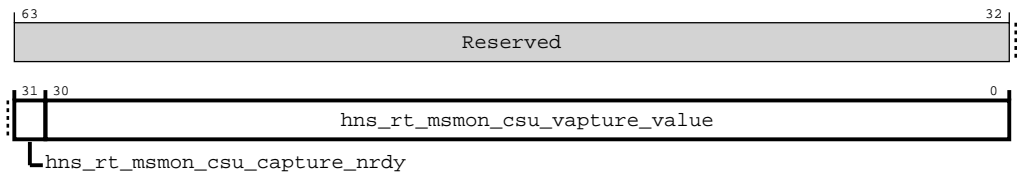


Table 5-79: cmn_hns_rt_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_msmon_csu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rt_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

5.2.2.65 cmn_hns_rt_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8860

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-76: cmn_hns_rt_msmon_mbwu

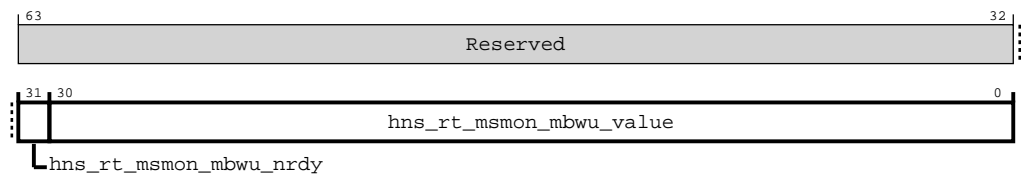


Table 5-80: cmn_hns_rt_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_msmon_mbwu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rt_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

5.2.2.66 cmn_hns_rt_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8868

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-77: cmn_hns_rt_msmon_mbwu_capture

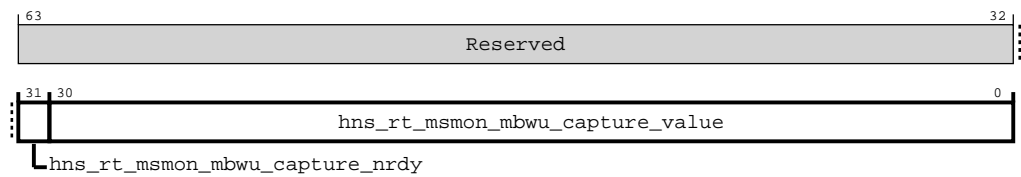


Table 5-81: cmn_hns_rt_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rt_msmon_mbwu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rt_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

5.2.2.67 cmn_hns_rt_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h9000

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-78: cmn_hns_rt_mpamcfg_cpbm

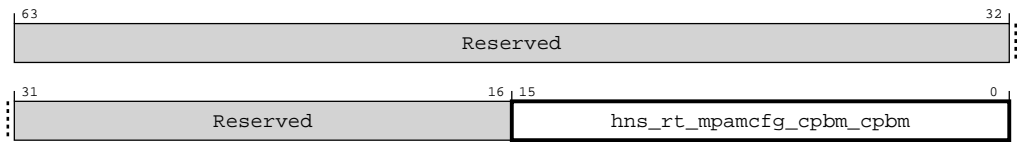


Table 5-82: cmn_hns_rt_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rt_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. NOTE: CPBM can not be all zeros for any PARTID.	RW	16'hFFFF

5.2.3 HN-S MPAM_NS register descriptions

This section lists the HN-S MPAM_NS registers.

5.2.3.1 cmn_hns_mpam_ns_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-79: cmn_hns_mpam_ns_node_info

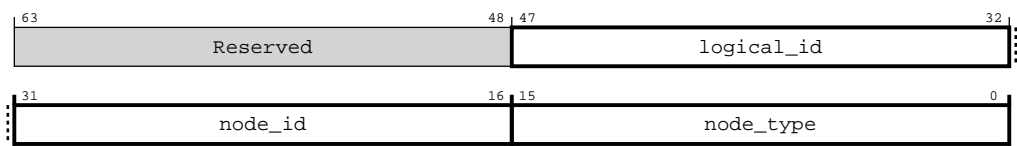


Table 5-83: cmn_hns_mpam_ns_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	\$logical_id_description	RO	Configuration dependent
[31:16]	node_id	\$node_id_description	RO	Configuration dependent
[15:0]	node_type	\$node_type_description	RO	Configuration dependent

5.2.3.2 cmn_hns_mpam_ns_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h80

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-80: cmn_hns_mpam_ns_child_info

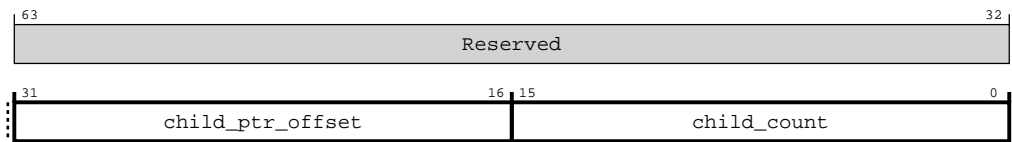


Table 5-84: cmn_hns_mpam_ns_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

5.2.3.3 cmn_hns_ns_mpam_idr

MPAM features ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1000

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-81: cmn_hns_ns_mpam_idr

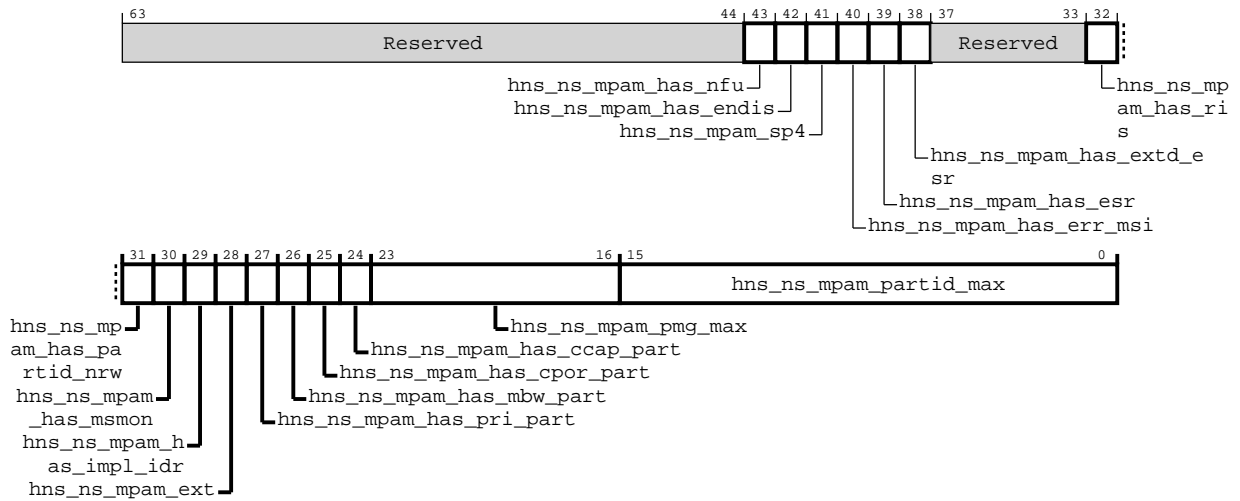


Table 5-85: cmn_hns_ns_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_ns_mpam_has_nfu	0: HN-F does not support no future use field 1: HN-F supports no future use field	RO	1'b0
[42]	hns_ns_mpam_has_endis	0: HN-F does not support PARTID enable and disable functionality 1: HN-F supports PARTID enable and disable functionality	RO	1'b0
[41]	hns_ns_mpam_sp4	0: HN-F supports two PARTID spaces 1: HN-F supports four PARTID spaces	RO	1'b1
[40]	hns_ns_mpam_has_err_msi	0: HN-F does not support MSI writes to signal MPAM error interrupt 1: HN-F supports MSI writes to signal MPAM error interrupt	RO	1'b0
[39]	hns_ns_mpam_has_esr	0: HN-F does not support MPAM error handling 1: HN-F supports MPAM error handling	RO	1'b1
[38]	hns_ns_mpam_has_extd_esr	0: MPAMF_ESR is 32 bits 1: MPAMF_ESR is 64 bits	RO	1'b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_ns_mpam_has_ris	0: HN-F does not support MPAM resource instance selector 1: HN-F supports MPAM resource instance selector	RO	1'b0
[31]	hns_ns_mpam_has_partid_nrw	0: HN-F does not support MPAM PARTID Narrowing 1: HN-F supports MPAM PARTID Narrowing	RO	Configuration dependent
[30]	hns_ns_mpam_has_msmon	0: MPAM performance monitoring is not supported 1: MPAM performance monitoring is supported	RO	Configuration dependent
[29]	hns_ns_mpam_has_impl_idr	0: MPAM implementation specific partitioning features not supported 1: MPAM implementation specific partitioning features supported	RO	Configuration dependent
[28]	hns_ns_mpam_ext	0: HN-F has no defined bits in [63:32] 1: HN-F has bits defined in [63:32]	RO	1'b1
[27]	hns_ns_mpam_has_pri_part	0: MPAM priority partitioning is not supported 1: MPAM priority partitioning is supported	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[26]	hns_ns_mpam_has_mbw_part	0: MPAM memory bandwidth partitioning is not supported 1: MPAM memory bandwidth partitioning is supported	RO	Configuration dependent
[25]	hns_ns_mpam_has_cpor_part	0: MPAM cache portion partitioning is not supported 1: MPAM cache portion partitioning is supported	RO	Configuration dependent
[24]	hns_ns_mpam_has_ccap_part	0: MPAM cache maximum capacity partitioning is not supported 1: MPAM cache maximum capacity partitioning is supported	RO	Configuration dependent
[23:16]	hns_ns_mpam_pmg_max	Maximum value of non-secure PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_ns_mpam_partid_max	Maximum value of non-secure PARTID supported by this HN-F	RO	Configuration dependent

5.2.3.4 cmn_hns_mpam_iidr

MPAM Implementation ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1018

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-82: cmn_hns_mpam_iidr

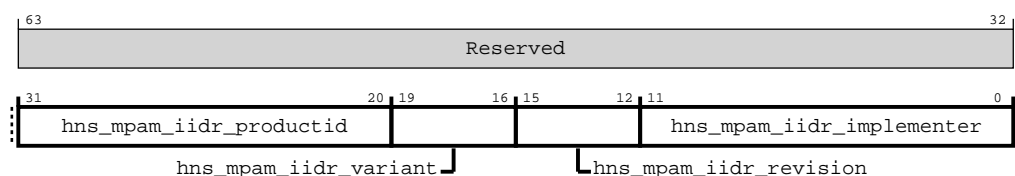


Table 5-86: cmn_hns_mpam_iidr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	hns_mpam_iidr_productid	Implementation defined value identifying MPAM memory system component	RO	12'h000
[19:16]	hns_mpam_iidr_variant	Implementation defined value identifying major revision of the product	RO	4'b0000
[15:12]	hns_mpam_iidr_revision	Implementation defined value identifying minor revision of the product	RO	4'b0000
[11:0]	hns_mpam_iidr_implementer	Implementation defined value identifying company that implemented MPAM memory system component	RO	12'h43B

5.2.3.5 cmn_hns_mpam_aidr

MPAM architecture ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1020

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-83: cmn_hns_mpam_aidr

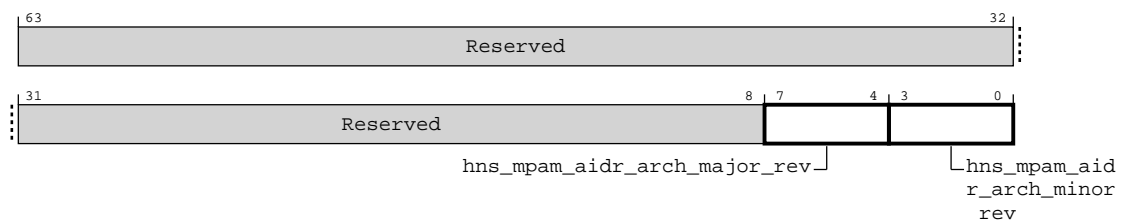


Table 5-87: cmn_hns_mpam_aidr attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	hns_mpam_aidr_arch_major_rev	Major revision of the MPAM architecture that this memory system component implements	RO	4'b0001
[3:0]	hns_mpam_aidr_arch_minor_rev	Minor revision of the MPAM architecture that this memory system component implements	RO	4'b0001

5.2.3.6 cmn_hns_ns_mpam_impl_idr

MPAM Implementation defined partitioning feature ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1028

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-84: cmn_hns_ns_mpam_impl_idr

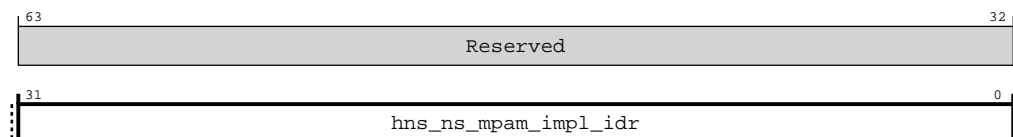


Table 5-88: cmn_hns_ns_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	hns_ns_mpam_impl_idr	Implementation defined partitioning features.	RO	32'h00000000

5.2.3.7 cmn_hns_ns_mpam_cpor_idr

MPAM cache portion partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1030

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-85: cmn_hns_ns_mpam_cpor_idr

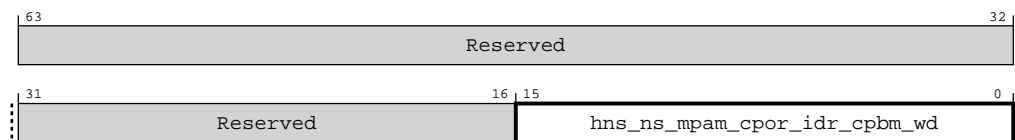


Table 5-89: cmn_hns_ns_mpam_cpor_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpam_cpor_idr_cpbm_wd	Number of bits in the cache portion partitioning bit map of this device.	RO	Configuration dependent

5.2.3.8 cmn_hns_ns_mpam_ccap_idr

MPAM cache capacity partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1038

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-86: cmn_hns_ns_mpam_ccap_idr

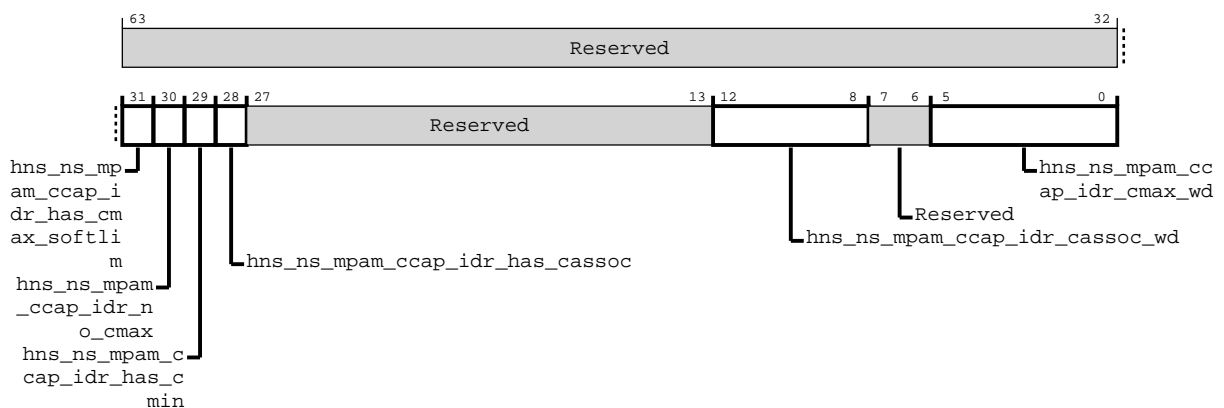


Table 5-90: cmn_hns_ns_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hns_ns_mpam_ccap_idr_has_cmax_softlim	0: HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit 1: HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit	RO	1'h0
[30]	hns_ns_mpam_ccap_idr_no_cmax	0: HN-F support MPAMCFG_CMAX 1: HN-F doesn't support MPAMCFG_CMAX	RO	1'h0
[29]	hns_ns_mpam_ccap_idr_has_cmin	0: HN-F does not support MPAMCFG_CMIN 1: HN-F supports MPAMCFG_CMIN	RO	1'h0
[28]	hns_ns_mpam_ccap_idr_has_cassoc	0: HN-F does not support MPAMCFG_CASSOC 1: HN-F supports MPAMCFG_CASSOC	RO	1'h0
[27:13]	Reserved	Reserved	RO	-
[12:8]	hns_ns_mpam_ccap_idr_cassoc_wd	Number of fractional bits implemented in the cache associativity partitioning.	RO	5'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	hns_ns_mpam_ccap_idr_cmax_wd	Number of fractional bits implemented in the cache capacity partitioning.	RO	Configuration dependent

5.2.3.9 cmn_hns_ns_mpam_mbw_idr

MPAM Memory Bandwidth partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1040

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-87: cmn_hns_ns_mpam_mbw_idr

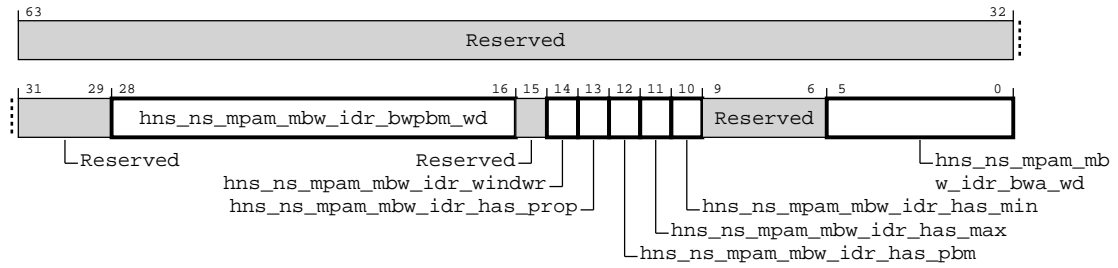


Table 5-91: cmn_hns_ns_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hns_ns_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register.	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hns_ns_mpam_mbw_idr_windwr	0: The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed. 1: The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.	RO	1'h0
[13]	hns_ns_mpam_mbw_idr_has_prop	0: There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register 1: MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.	RO	1'h0
[12]	hns_ns_mpam_mbw_idr_has_pbm	0: There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register 1: MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.	RO	1'h0
[11]	hns_ns_mpam_mbw_idr_has_max	0: There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register 1: MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.	RO	1'h0
[10]	hns_ns_mpam_mbw_idr_has_min	0: There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register 1: MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.	RO	1'h0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_ns_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

5.2.3.10 cmn_hns_ns_mpam_pri_idr

MPAM Priority partitioning ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1048

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-88: cmn_hns_ns_mpam_pri_idr

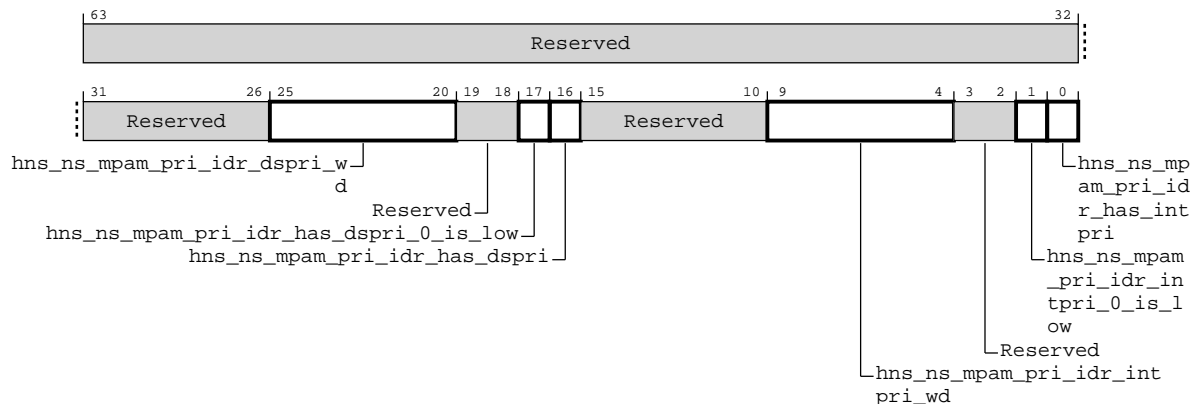


Table 5-92: cmn_hns_ns_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	<code>hns_ns_mpam_pri_idr_dspri_wd</code>	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	6'h0
[19:18]	Reserved	Reserved	RO	-
[17]	<code>hns_ns_mpam_pri_idr_has_dspri_0_is_low</code>	0: In the DSPRI field, a value of 0 means highest priority. 1: In the DSPRI field, a value of 0 means lowest priority.	RO	1'h0
[16]	<code>hns_ns_mpam_pri_idr_has_dspri</code>	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]	<code>hns_ns_mpam_pri_idr_intpri_wd</code>	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	6'h0

Bits	Name	Description	Type	Reset
[3:2]	Reserved	Reserved	RO	-
[1]	hns_ns_mpam_pri_idr_intpri_0_is_low	0: In the INTPRI field, a value of 0 means highest priority. 1: In the INTPRI field, a value of 0 means lowest priority.	RO	1'h0
[0]	hns_ns_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

5.2.3.11 cmn_hns_ns_mpam_partid_nrw_idr

MPAM PARTID narrowing ID register. This is a shared register for S and NS

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1050

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-89: cmn_hns_ns_mpam_partid_nrw_idr

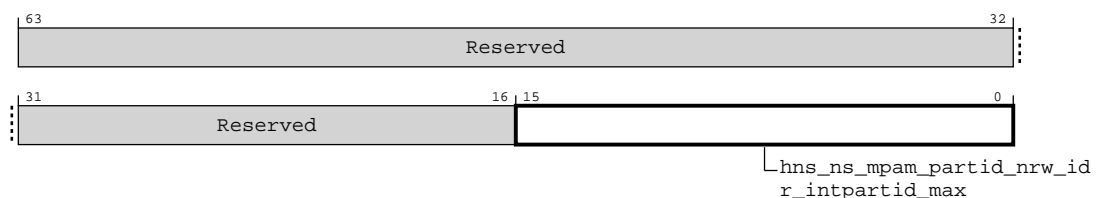


Table 5-93: cmn_hns_ns_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

5.2.3.12 cmn_hns_ns_mpam_msmon_idr

MPAM performance monitoring ID register. This is a shared register for S and NS.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1080

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-90: cmn_hns_ns_mpam_msmon_idr

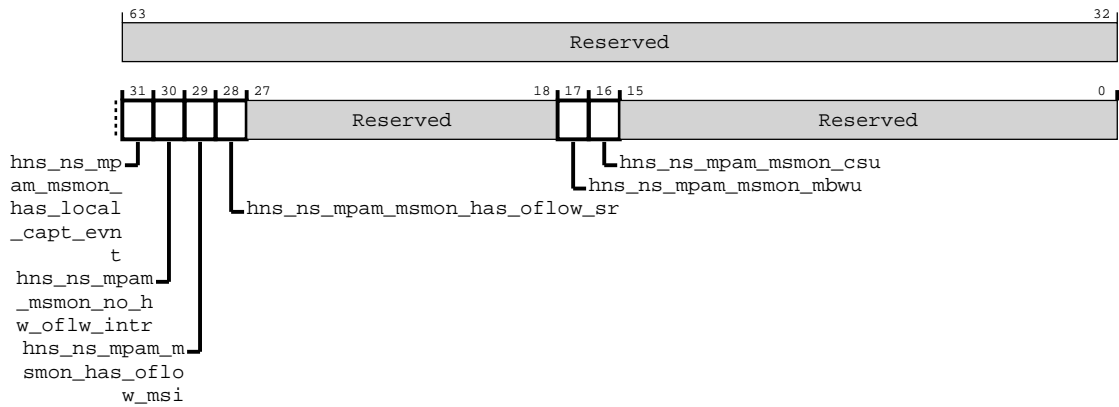


Table 5-94: cmn_hns_ns_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_ns_mpam_msmon_has_local_capt_evt</code>	Has the local capture event generator and the MSMON_CAPT_EVT register.	RO	1'h1
[30]	<code>hns_ns_mpam_msmon_no_hw_oflw_intr</code>	0: HNF does not have hardwired MPAM overflow interrupt 1: HNF has have hardwired MPAM overflow interrupt	RO	1'b0
[29]	<code>hns_ns_mpam_msmon_has_oflow_msi</code>	0: HNF does not have support for MSI writes to signal MPAM monitor overflow interrupt 1: HNF has support for MSI writes to signal MPAM monitor overflow interrupt	RO	1'b0
[28]	<code>hns_ns_mpam_msmon_has_oflow_sr</code>	0: HNF does not have overflow status register 1: HNF has overflow status register	RO	1'b0
[27:18]	Reserved	Reserved	RO	-
[17]	<code>hns_ns_mpam_msmon_mbwu</code>	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	<code>hns_ns_mpam_msmon_csu</code>	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

5.2.3.13 cmn_hns_ns_mpam_csumon_idr

MPAM cache storage usage monitor ID register. This is banked separately.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1088

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-91: cmn_hns_ns_mpam_csumon_idr

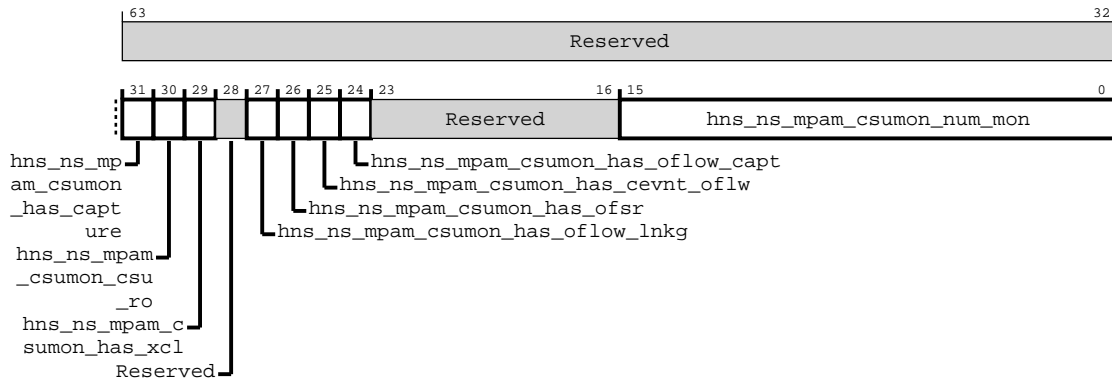


Table 5-95: cmn_hns_ns_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_ns_mpam_csumon_has_capture</code>	0: MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature. 1: This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.	RO	1'h1
[30]	<code>hns_ns_mpam_csumon_csu_ro</code>	0: MSMON_CSU is read/write. 1: MSMON_CSU is read-only.	RO	1'b0
[29]	<code>hns_ns_mpam_csumon_has_xcl</code>	0: MSMON_CFG_CSU_FLT does not implement the XCL field 1: MSMON_CFG_CSU_FLT implements the XCL field	RO	1'b0
[28]	Reserved	Reserved	RO	-
[27]	<code>hns_ns_mpam_csumon_has_oflow_lnk</code>	0: HNF does not support CSU overflow linkage 1: HNF supports CSU overflow linkage	RO	1'b0
[26]	<code>hns_ns_mpam_csumon_has_ofsr</code>	0: MSMON_CSU_OFSR register is not implemented 1: MSMON_CSU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_ns_mpam_csumon_has_cevnt_oflw</code>	0: HNF does not support MSMON_CFG_CSU_CTL.CEVNT_OFLW 1: HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW	RO	1'b0

Bits	Name	Description	Type	Reset
[24]	hns_ns_mpam_csumon_has_oflow_capt	0: HNF does not support MSMON_CFG_CSU_CTL.OFLOW_CAPT 1: HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT	RO	1'b0
[23:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	Configuration dependent

5.2.3.14 cmn_hns_ns_mpam_mbwumon_idr

MPAM memory bandwidth usage monitor ID register. This is a shared register for S and NS.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1090

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-92: cmn_hns_ns_mpam_mbwumon_idr

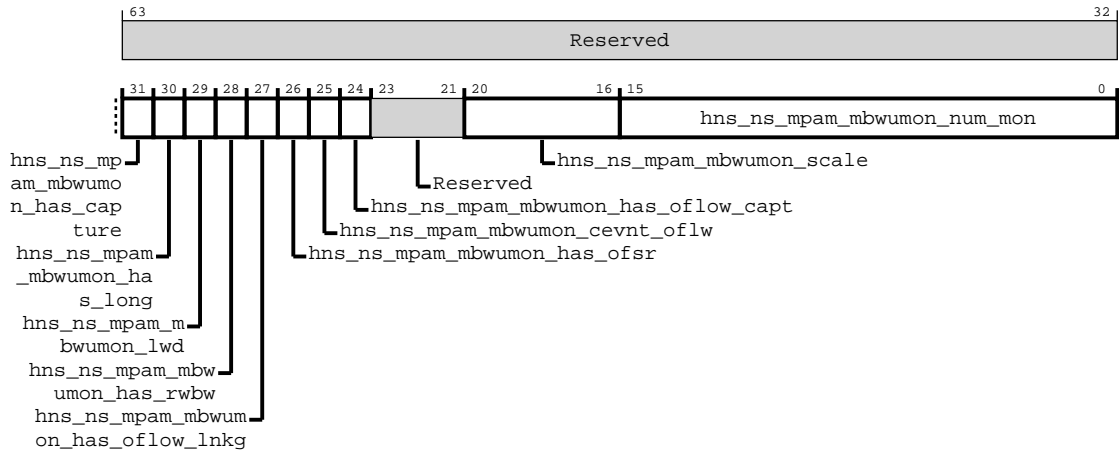


Table 5-96: cmn_hns_ns_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_ns_mpam_mbwumon_has_capture</code>	0: MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature. 1: This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behavior.	RO	1'h0
[30]	<code>hns_ns_mpam_mbwumon_has_long</code>	0: MSMON_MBWU_L is not implemented. 1: MSMON_MBWU_L is implemented.	RO	1'b0
[29]	<code>hns_ns_mpam_mbwumon_lwd</code>	0: MSMON_MBWU_L has 44-bit VALUE field in bits [43:0]. 1: MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].	RO	1'b0
[28]	<code>hns_ns_mpam_mbwumon_has_rwbw</code>	0: Read/write bandwidth selection is not implemented. 1: Read/write bandwidth selection is implemented.	RO	1'b0
[27]	<code>hns_ns_mpam_mbwumon_has_oflow_lnkg</code>	0: Does not support MSMON_CFG_MBWU_CTL.OFLOW_LNKG. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.	RO	1'b0
[26]	<code>hns_ns_mpam_mbwumon_has_ofsr</code>	0: MSMON_MBWU_OFSR register is not implemented 1: MSMON_MBWU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_ns_mpam_mbwumon_cevnt_oflw</code>	0: Does not support MSMON_CFG_MBWU_CTL.CEVNT_OFLW. 1: Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.	RO	1'b0
[24]	<code>hns_ns_mpam_mbwumon_has_oflow_capt</code>	0: Does not support MSMON_CFG_MBWU_CTL.OFLOW_CAPT. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.	RO	1'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	<code>hns_ns_mpam_mbwumon_scale</code>	Scaling of MSMON_MBWU.VALUE in bits.	RO	5'h0
[15:0]	<code>hns_ns_mpam_mbwumon_num_mon</code>	The number of MBWU monitoring counters implemented in this component.	RO	16'h0

5.2.3.15 cmn_hns_ns_mpam_ecr

MPAM Error Control Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h10F0

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-93: cmn_hns_ns_mpam_ecr

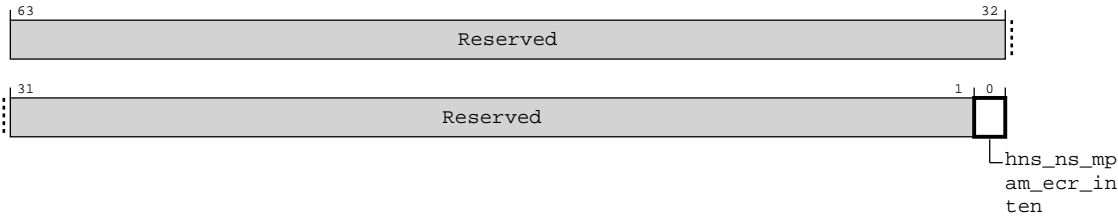


Table 5-97: cmn_hns_ns_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hns_ns_mpam_ecr_inten	Interrupt Enable. When INTEN = 0, MPAM error interrupts are not generated. When INTEN = 1, MPAM error interrupts are generated.	RW	1'h0

5.2.3.16 cmn_hns_ns_mpam_esr

MPAM Error Status Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h10F8

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-94: cmn_hns_ns_mpam_esr

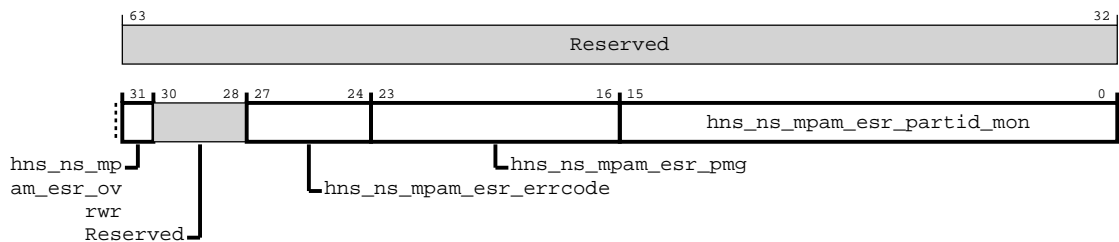


Table 5-98: cmn_hns_ns_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpam_esr_ovrwr	Overwritten. If 0 and ERRCODE is zero, no errors have occurred. If 0 and ERRCODE is non-zero, a single error has occurred and is recorded in this register. If 1 and ERRCODE is non-zero, multiple errors have occurred and this register records the most recent error. The state where this bit is 1 and ERRCODE is zero is not produced by hardware and is only reached when software writes this combination into this register.	RW	1'h0
[30:28]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[27:24]	hns_ns_mpam_esr_errcode	Error code	RW	4'h0
[23:16]	hns_ns_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hns_ns_mpam_esr_partid_mon	PARTID captured if the error code captures PARTID. MON selector captured if the error code captures MON. Otherwise 0x0000.	RW	16'h0

5.2.3.17 cmn_hns_ns_mpamcfg_part_sel

MPAM partition configuration selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1100

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-95: cmn_hns_ns_mpamcfg_part_sel

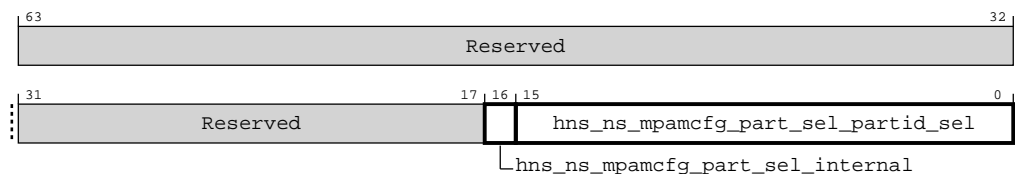


Table 5-99: cmn_hns_ns_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[16]	hns_ns_mpamcfg_part_sel_internal	If MPAMF_IDR.HAS_PARTID_NRW = 0, this field is RAZ/WI. If MPAMF_IDR.HAS_PARTID_NRW = 1, this bit decides how to interpret PARTID_SEL.	RW	1'h0
[15:0]	hns_ns_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

5.2.3.18 cmn_hns_ns_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1108

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-96: cmn_hns_ns_mpamcfg_cmax

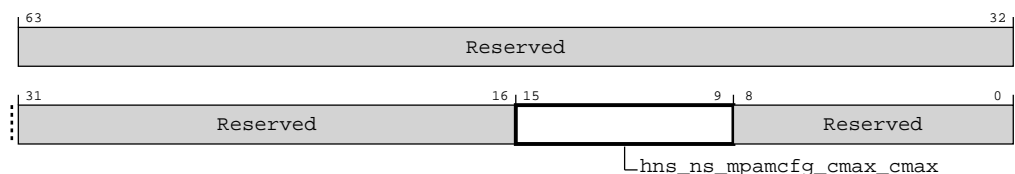


Table 5-100: cmn_hns_ns_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:9]	hns_ns_mpamcfg_cmax_cmax	Maximum cache capacity usage in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL.	RW	7'b11111111
[8:0]	Reserved	Reserved	RO	-

5.2.3.19 cmn_hns_ns_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1200

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-97: cmn_hns_ns_mpamcfg_mbw_min

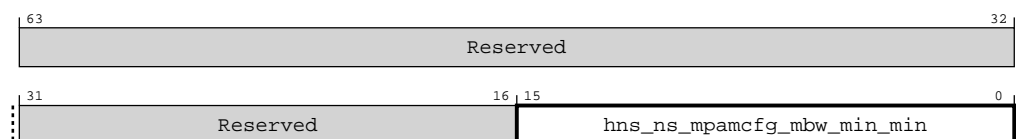


Table 5-101: cmn_hns_ns_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpamcfg_mbw_min_min	Memory minimum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.20 cmn_hns_ns_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1208

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-98: cmn_hns_ns_mpamcfg_mbw_max

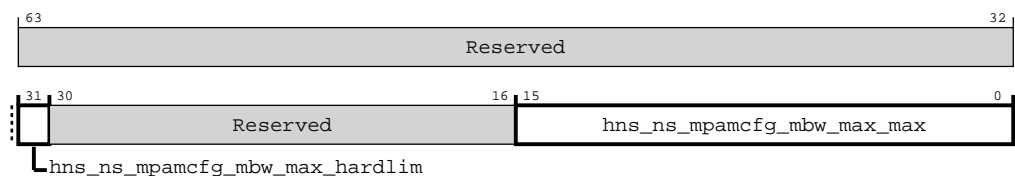


Table 5-102: cmn_hns_ns_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_ns_mpamcfg_mbw_max_hardlim</code>	0: When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth. 1: When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_ns_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.21 cmn_hns_ns_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1220

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-99: cmn_hns_ns_mpamcfg_mbw_winwd

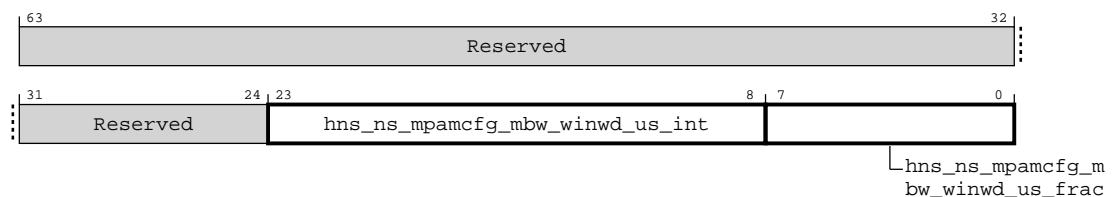


Table 5-103: cmn_hns_ns_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hns_ns_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hns_ns_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

5.2.3.22 cmn_hns_ns_mpamcfg_pri

MPAM priority partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1400

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-100: cmn_hns_ns_mpamcfg_pri

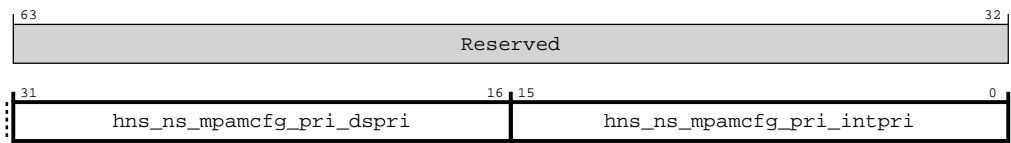


Table 5-104: cmn_hns_ns_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	hns_ns_mpamcfg_pri_dspri	If HAS_DSPRI is 1, this field is a priority value applied to downstream communications from this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	16'h0
[15:0]	hns_ns_mpamcfg_pri_intpri	If HAS_INTPRI is 1, this field is a priority value applied internally inside this memory system component for transactions of the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.23 cmn_hns_ns_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1500

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-101: cmn_hns_ns_mpamcfg_mbw_prop

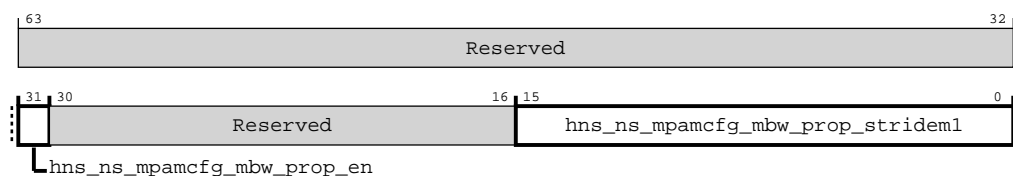


Table 5-105: cmn_hns_ns_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_mpamcfg_mbw_prop_en	0: The selected partition is not regulated by proportional stride bandwidth partitioning. 1: The selected partition has bandwidth usage regulated by proportional stride bandwidth partitioning as controlled by STRIDEM1.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_mpamcfg_mbw_prop_stridem1	Normalized cost of a bandwidth consumption by the partition. STRIDEM1 is the stride for the partition minus one.	RW	16'h0

5.2.3.24 cmn_hns_ns_mpamcfg_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1600

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-102: cmn_hns_ns_mpamcfg_intpartid

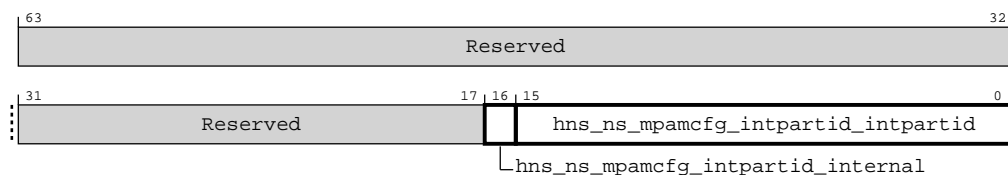


Table 5-106: cmn_hns_ns_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	<code>hns_ns_mpamcfg_intpartid_internal</code>	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]	<code>hns_ns_mpamcfg_intpartid_intpartid</code>	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.25 cmn_hns_ns_msmon_cfg_mon_sel

Memory system performance monitor selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1800

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-103: cmn_hns_ns_msmon_cfg_mon_sel

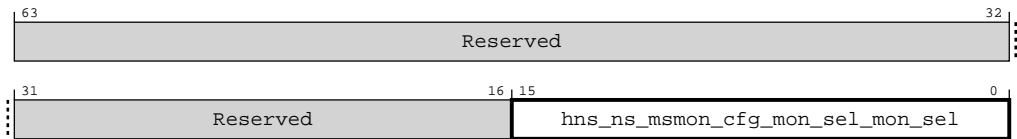


Table 5-107: cmn_hns_ns_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_ns_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure.	RW	16'h0

5.2.3.26 cmn_hns_ns_msmon_capt_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1808

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-104: cmn_hns_ns_msmon_capt_evnt

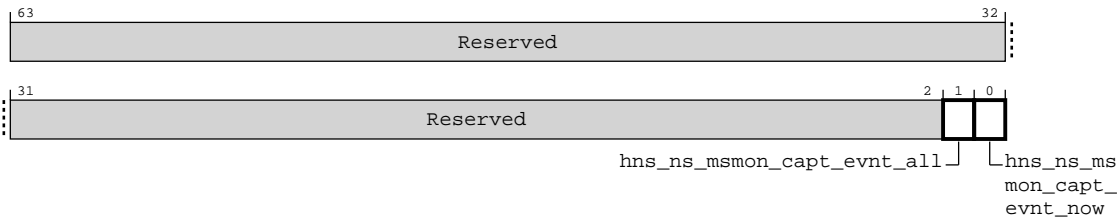


Table 5-108: cmn_hns_ns_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1]	hns_ns_msmon_capt_evnt_all	In secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to secure monitors in this memory system component with CAPT_EVNT = 7. In non-secure version if NOW is written as 1, signal a capture event to non-secure monitors in this memory system component with CAPT_EVNT = 7. In root version, if ALL written as 1 and NOW is also written as 1, signal a capture event to root, realm, secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to root monitors in this memory system component with CAPT_EVNT = 7. In realm version, if ALL written as 1 and NOW is also written as 1, signal a capture event to realm and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to realm monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hns_ns_msmon_capt_evnt_now	When written as 1, this bit causes an event to all monitors in this memory system component with CAPT_EVNT set to the value of 7. When this bit is written as 0, no event is signalled.	RW	1'h0

5.2.3.27 cmn_hns_ns_msmon_cfg_csuflt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1810

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-105: cmn_hns_ns_msmon_cfg_csu_flt

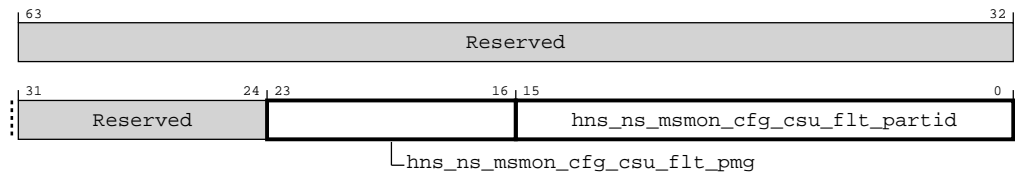


Table 5-109: cmn_hns_ns_msmon_cfg_csu_flt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_ns_msmon_cfg_csu_flt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_ns_msmon_cfg_csu_flt_partid	Configures the cache storage usage performance monitor to a PARTID. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures the storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	16'h0

5.2.3.28 cmn_hns_ns_msmon_cfg_csu_ctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1818

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-106: cmn_hns_ns_msmon_cfg_csu_ctl

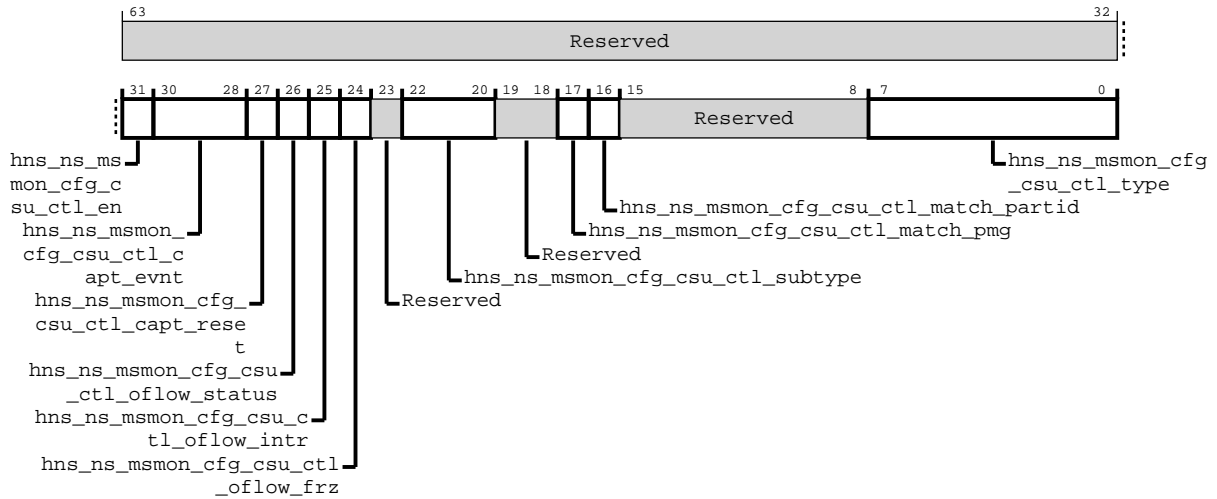


Table 5-110: cmn_hns_ns_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_ns_msmon_cfg_csu_ctl_en</code>	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]	<code>hns_ns_msmon_cfg_csu_ctl_capt_evnt</code>	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]	<code>hns_ns_msmon_cfg_csu_ctl_capt_reset</code>	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	<code>hns_ns_msmon_cfg_csu_ctl_oflow_status</code>	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	<code>hns_ns_msmon_cfg_csu_ctl_oflow_intr</code>	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	<code>hns_ns_msmon_cfg_csu_ctl_oflow_frz</code>	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]	Reserved	Reserved	RO	-
[22:20]	<code>hns_ns_msmon_cfg_csu_ctl_subtype</code>	Not currently used for CSU monitors, but reserved for future use.	RW	3'h0
[19:18]	Reserved	Reserved	RO	-
[17]	<code>hns_ns_msmon_cfg_csu_ctl_match_pmg</code>	0: Monitor storage used by all PMG values. 1: Only monitor storage used with the PMG value matching <code>MSMON_CFG_CSU_FLT.PMG</code> .	RW	1'h0
[16]	<code>hns_ns_msmon_cfg_csu_ctl_match_partid</code>	0: Monitor storage used by all PARTIDs. 1: Only monitor storage used with the PARTID matching <code>MSMON_CFG_CSU_FLT.PARTID</code> .	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	<code>hns_ns_msmon_cfg_csu_ctl_type</code>	Read-only: Constant type indicating the type of the monitor. CSU monitor is <code>TYPE = 0x43</code> .	RW	8'h43

5.2.3.29 cmn_hns_ns_msmon_cfg_mbwuflt

Memory system performance monitor configure memory bandwidth usage monitor filter register.
This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1820

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-107: cmn_hns_ns_msmon_cfg_mbwuflt

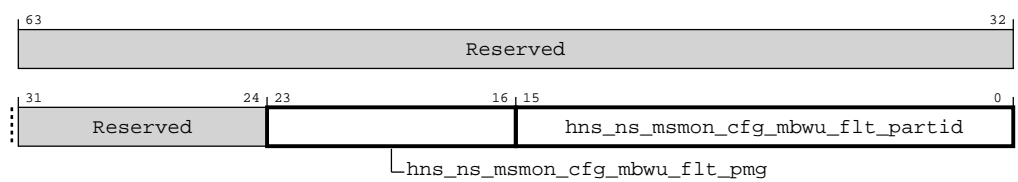


Table 5-111: cmn_hns_ns_msmon_cfg_mbwuflt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_ns_msmon_cfg_mbwuflt_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

Bits	Name	Description	Type	Reset
[15:0]	hns_ns_msmon_cfg_mbwuflt_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

5.2.3.30 cmn_hns_ns_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1828

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-108: cmn_hns_ns_msmon_cfg_mbwu_ctl

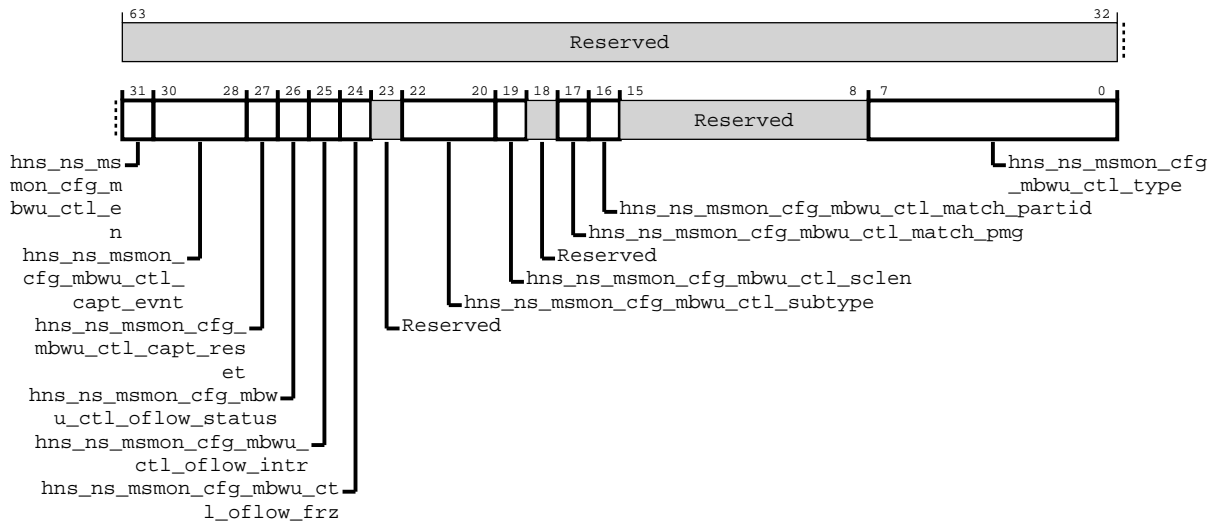


Table 5-112: cmn_hns_ns_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_cfg_mbwu_ctl_en	0: The monitor is disabled and must not collect any information. 1: The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_ns_msmon_cfg_mbwu_ctl_capt_evnt	Select the event that triggers capture from the following: 0: No capture event is triggered. 1: External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_ns_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hns_ns_msmon_cfg_mbwu_ctl_oflow_status	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_ns_msmon_cfg_mbwu_ctl_oflow_intr	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_ns_msmon_cfg_mbwu_ctl_oflow_frz	0: Monitor count wraps on overflow. 1: Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.	RW	1'h0
[23]	Reserved	Reserved	RO	-
[22:20]	hns_ns_msmon_cfg_mbwu_ctl_subtype	A monitor can have other event matching criteria. The meaning of values in this field varies by monitor type. The MBWU monitor type supports: 0: Do not count any bandwidth. 1: Count bandwidth used by memory reads 2: Count bandwidth used by memory writes 3: Count bandwidth used by memory reads and memory writes All other values are reserved and behaviour of a monitor with SUBTYPE set to one of the reserved values is UNPREDICTABLE.	RW	3'h0
[19]	hns_ns_msmon_cfg_mbwu_ctl_sclen	0: MSMON_MBWU.VALUE has bytes counted by the monitor instance. 1: MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.	RW	1'h0
[18]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[17]	hns_ns_msmon_cfg_mbwu_ctl_match_pmg	0: Monitor bandwidth used by all PMG values. 1: Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_ns_msmon_cfg_mbwu_ctl_match_partid	0: Monitor bandwidth used by all PARTIDs. 1: Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_ns_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

5.2.3.31 cmn_hns_ns_msmon_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1840

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-109: cmn_hns_ns_msmon_csu

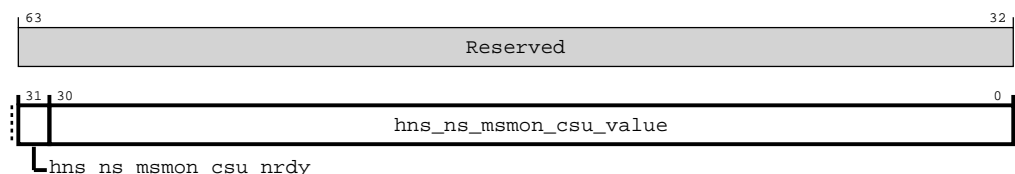


Table 5-113: cmn_hns_ns_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_csu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_csu_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

5.2.3.32 cmn_hns_ns_msmon_csu_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1848

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-110: cmn_hns_ns_msmon_csu_capture

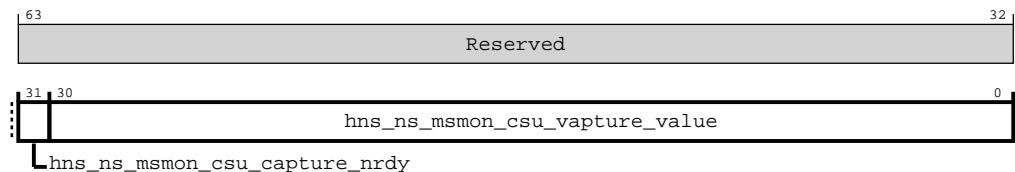


Table 5-114: cmn_hns_ns_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_csu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_csu_vapture_value	Cache storage usage value if NRDY is 0. Invalid if NRDY is 1. VALUE is the cache storage usage in bytes.	RW	31'h0

5.2.3.33 cmn_hns_ns_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1860

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-111: cmn_hns_ns_msmon_mbwu

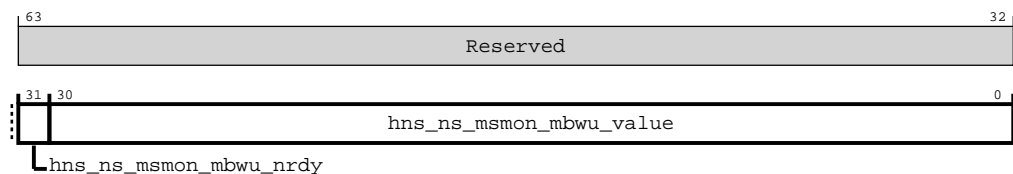


Table 5-115: cmn_hns_ns_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_mbwu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_mbwu_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

5.2.3.34 cmn_hns_ns_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1868

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-112: cmn_hns_ns_msmon_mbwu_capture

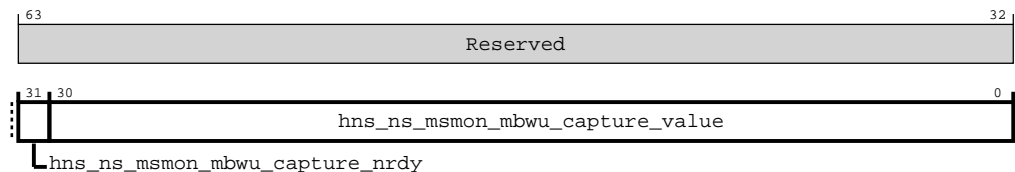


Table 5-116: cmn_hns_ns_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_ns_msmon_mbwu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_ns_msmon_mbwu_capture_value	Memory channel bandwidth value if NRDY is 0. Invalid if NRDY is 1. VALUE is the memory channel bandwidth usage in megabytes.	RW	31'h0

5.2.3.35 cmn_hns_ns_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h2000

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-113: cmn_hns_ns_mpamcfg_cpbm

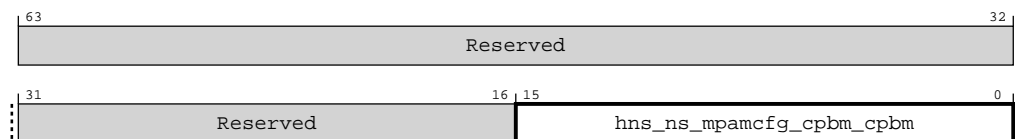


Table 5-117: cmn_hns_ns_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_ns_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. NOTE: CPBM can not be all zeros for any PARTID.	RW	16'hFFFF

5.2.3.36 cmn_hns_rl_mpam_idr

MPAM features ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8000

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-114: cmn_hns_rl_mpam_idr

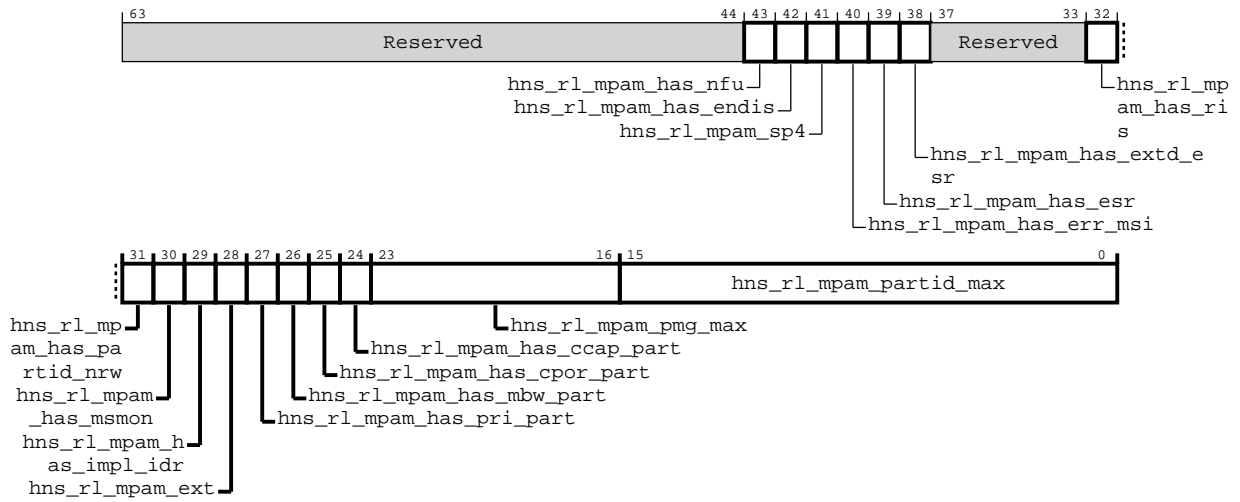


Table 5-118: cmn_hns_rl_mpam_idr attributes

Bits	Name	Description	Type	Reset
[63:44]	Reserved	Reserved	RO	-
[43]	hns_rl_mpam_has_nfu	0: HN-F does not support no future use field 1: HN-F supports no future use field	RO	1'b0
[42]	hns_rl_mpam_has_endis	0: HN-F does not support PARTID enable and disable functionality 1: HN-F supports PARTID enable and disable functionality	RO	1'b0
[41]	hns_rl_mpam_sp4	0: HN-F supports two PARTID spaces 1: HN-F supports four PARTID spaces	RO	1'b1
[40]	hns_rl_mpam_has_err_msi	0: HN-F does not support MSI writes to signal MPAM error interrupt 1: HN-F supports MSI writes to signal MPAM error interrupt	RO	1'b0
[39]	hns_rl_mpam_has_esr	0: HN-F does not support MPAM error handling 1: HN-F supports MPAM error handling	RO	1'b1
[38]	hns_rl_mpam_has_extd_esr	0: MPAMF_ESR is 32 bits 1: MPAMF_ESR is 64 bits	RO	1'b0
[37:33]	Reserved	Reserved	RO	-
[32]	hns_rl_mpam_has_ris	0: HN-F does not support MPAM resource instance selector 1: HN-F supports MPAM resource instance selector	RO	1'b0
[31]	hns_rl_mpam_has_partid_nrw	0: HN-F does not support MPAM PARTID Narrowing 1: HN-F supports MPAM PARTID Narrowing	RO	Configuration dependent
[30]	hns_rl_mpam_has_msmon	0: MPAM performance monitoring is not supported 1: MPAM performance monitoring is supported	RO	Configuration dependent
[29]	hns_rl_mpam_has_impl_idr	0: MPAM implementation specific partitioning features not supported 1: MPAM implementation specific partitioning features supported	RO	Configuration dependent
[28]	hns_rl_mpam_ext	0: HN-F has no defined bits in [63:32] 1: HN-F has bits defined in [63:32]	RO	1'b1
[27]	hns_rl_mpam_has_pri_part	0: MPAM priority partitioning is not supported 1: MPAM priority partitioning is supported	RO	Configuration dependent
[26]	hns_rl_mpam_has_mbw_part	0: MPAM memory bandwidth partitioning is not supported 1: MPAM memory bandwidth partitioning is supported	RO	Configuration dependent
[25]	hns_rl_mpam_has_cpor_part	0: MPAM cache portion partitioning is not supported 1: MPAM cache portion partitioning is supported	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[24]	hns_rl_mpam_has_ccap_part	0: MPAM cache maximum capacity partitioning is not supported 1: MPAM cache maximum capacity partitioning is supported	RO	Configuration dependent
[23:16]	hns_rl_mpam_pmg_max	Maximum value of realm PMG supported by this HN-F	RO	Configuration dependent
[15:0]	hns_rl_mpam_partid_max	Maximum value of realm PARTID supported by this HN-F	RO	Configuration dependent

5.2.3.37 cmn_hns_rl_mpam_impl_idr

MPAM Implementation defined partitioning feature ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8028

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-115: cmn_hns_rl_mpam_impl_idr

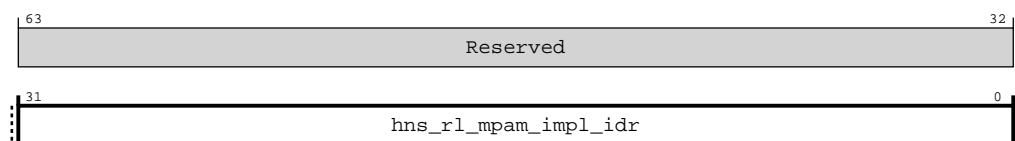


Table 5-119: cmn_hns_rl_mpam_impl_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

5.2.3.39 cmn_hns_rl_mpam_ccap_idr

MPAM cache capacity partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8038

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-117: cmn_hns_rl_mpam_ccap_idr

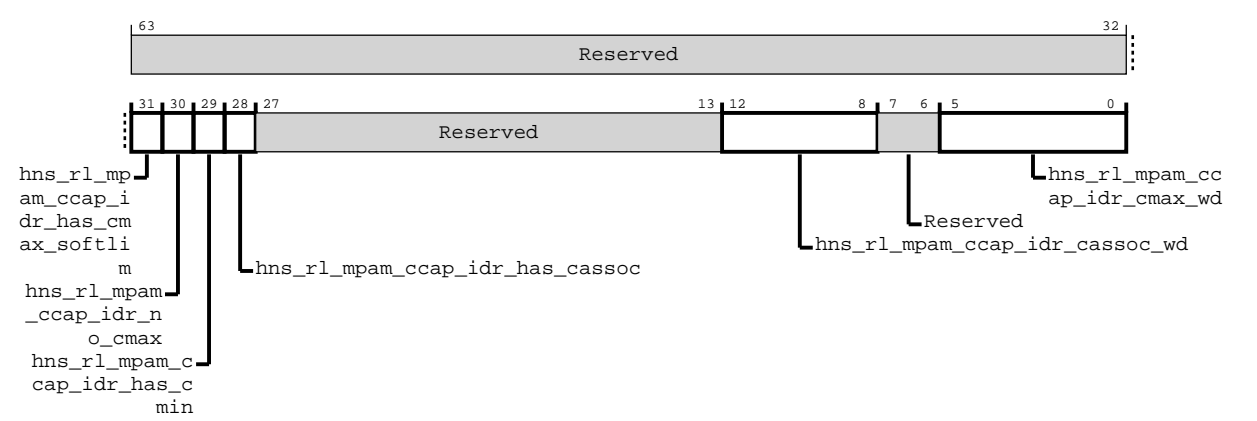


Table 5-121: cmn_hns_rl_mpam_ccap_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31]	hns_rl_mpam_ccap_idr_has_cmax_softlim	0: HN-F has no SOFTLIM field and the maximum capacity is controlled with a hard limit 1: HN-F has a SOFTLIM field and the maximum capacity is controlled with a hard limit	RO	1'h0
[30]	hns_rl_mpam_ccap_idr_no_cmax	0: HN-F support MPAMCFG_CMAX 1: HN-F doesn't support MPAMCFG_CMAX	RO	1'h0
[29]	hns_rl_mpam_ccap_idr_has_cmin	0: HN-F does not support MPAMCFG_CMIN 1: HN-F supports MPAMCFG_CMIN	RO	1'h0
[28]	hns_rl_mpam_ccap_idr_has_cassoc	0: HN-F does not support MPAMCFG_CASSOC 1: HN-F supports MPAMCFG_CASSOC	RO	1'h0
[27:13]	Reserved	Reserved	RO	-
[12:8]	hns_rl_mpam_ccap_idr_cassoc_wd	Number of fractional bits implemented in the cache associativity partitioning.	RO	5'h0
[7:6]	Reserved	Reserved	RO	-
[5:0]	hns_rl_mpam_ccap_idr_cmax_wd	Number of fractional bits implemented in the cache capacity partitioning.	RO	Configuration dependent

5.2.3.40 cmn_hns_rl_mpam_mbw_idr

MPAM Memory Bandwidth partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8040

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-118: cmn_hns_rl_mpam_mbw_idr

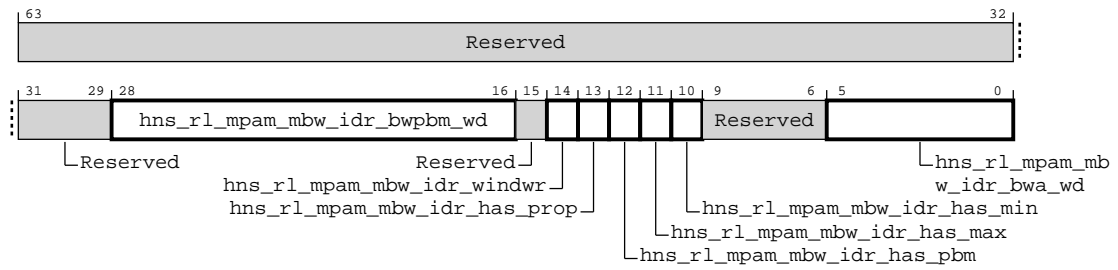


Table 5-122: cmn_hns_rl_mpam_mbw_idr attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:16]	hns_rl_mpam_mbw_idr_bwpbm_wd	Number of bits indication portions in MPAMCFG_MBW_PBM register.	RO	13'h0
[15]	Reserved	Reserved	RO	-
[14]	hns_rl_mpam_mbw_idr_windwr	0: The bandwidth accounting period should be read from MPAMCFG_MBW_WINDWR register, which might be fixed. 1: The bandwidth accounting width is readable and writable per partition in MPAMCFG_MBW_WINDWR register.	RO	1'h0
[13]	hns_rl_mpam_mbw_idr_has_prop	0: There is no memory bandwidth proportional stride control and no MPAMCFG_MBW_PROP register 1: MPAMCFG_MBW_PROP register exists and memory bandwidth proportional stride memory bandwidth allocation scheme is supported.	RO	1'h0
[12]	hns_rl_mpam_mbw_idr_has_pbm	0: There is no memory bandwidth portion control and no MPAMCFG_MBW_PBM register 1: MPAMCFG_MBW_PBM register exists and memory bandwidth portion allocation scheme is supported.	RO	1'h0
[11]	hns_rl_mpam_mbw_idr_has_max	0: There is no maximum memory bandwidth control and no MPAMCFG_MBW_MAX register 1: MPAMCFG_MBW_MAX register exists and maximum memory bandwidth allocation scheme is supported.	RO	1'h0
[10]	hns_rl_mpam_mbw_idr_has_min	0: There is no minimum memory bandwidth control and no MPAMCFG_MBW_MIN register 1: MPAMCFG_MBW_MIN register exists and minimum memory bandwidth allocation scheme is supported.	RO	1'h0
[9:6]	Reserved	Reserved	RO	-
[5:0]	hns_rl_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

5.2.3.41 cmn_hns_rl_mpam_pri_idr

MPAM Priority partitioning ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8048

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-119: cmn_hns_rl_mpam_pri_idr

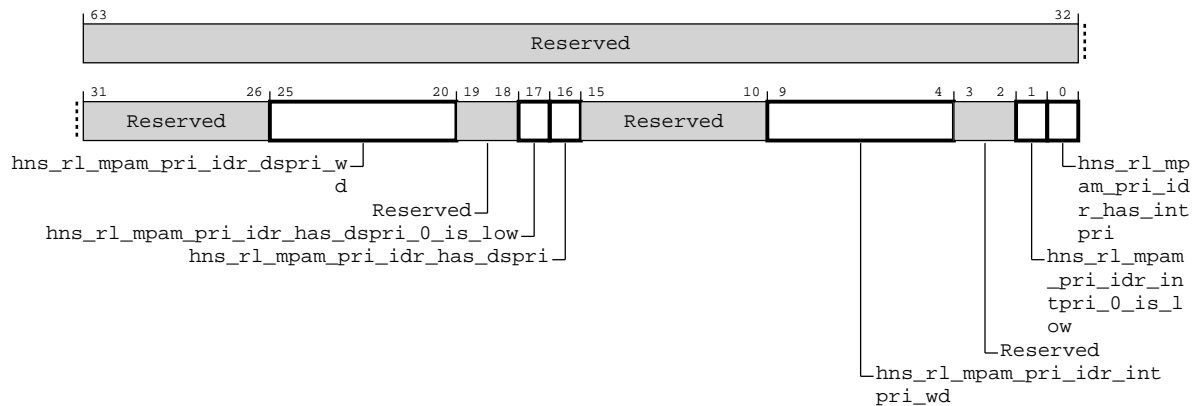


Table 5-123: cmn_hns_rl_mpam_pri_idr attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:20]	<code>hns_rl_mpam_pri_idr_dspri_wd</code>	Number of bits in downstream priority field (DSPRI) in MPAMCFG_PRI.	RO	6'h0
[19:18]	Reserved	Reserved	RO	-
[17]	<code>hns_rl_mpam_pri_idr_has_dspri_0_is_low</code>	0: In the DSPRI field, a value of 0 means highest priority. 1: In the DSPRI field, a value of 0 means lowest priority.	RO	1'h0
[16]	<code>hns_rl_mpam_pri_idr_has_dspri</code>	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]	<code>hns_rl_mpam_pri_idr_intpri_wd</code>	Number of bits in the internal priority field (INTPRI) in MPAMCFG_PRI.	RO	6'h0

Bits	Name	Description	Type	Reset
[3:2]	Reserved	Reserved	RO	-
[1]	hns_rl_mpam_pri_idr_intpri_O_is_low	0: In the INTPRI field, a value of 0 means highest priority. 1: In the INTPRI field, a value of 0 means lowest priority.	RO	1'h0
[0]	hns_rl_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

5.2.3.42 cmn_hns_rl_mpam_partid_nrw_idr

MPAM PARTID narrowing ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8050

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-120: cmn_hns_rl_mpam_partid_nrw_idr

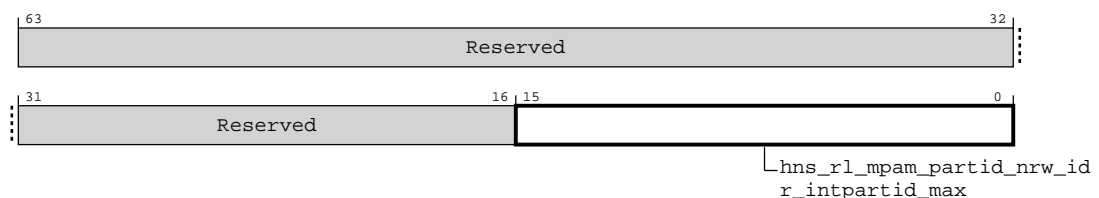


Table 5-124: cmn_hns_rl_mpam_partid_nrw_idr attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_mpam_partid_nrw_idr_intpartid_max	This field indicates the largest intPARTID supported in this component.	RO	16'h00

5.2.3.43 cmn_hns_rl_mpam_msmon_idr

MPAM performance monitoring ID register. This is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8080

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-121: cmn_hns_rl_mpam_msmon_idr

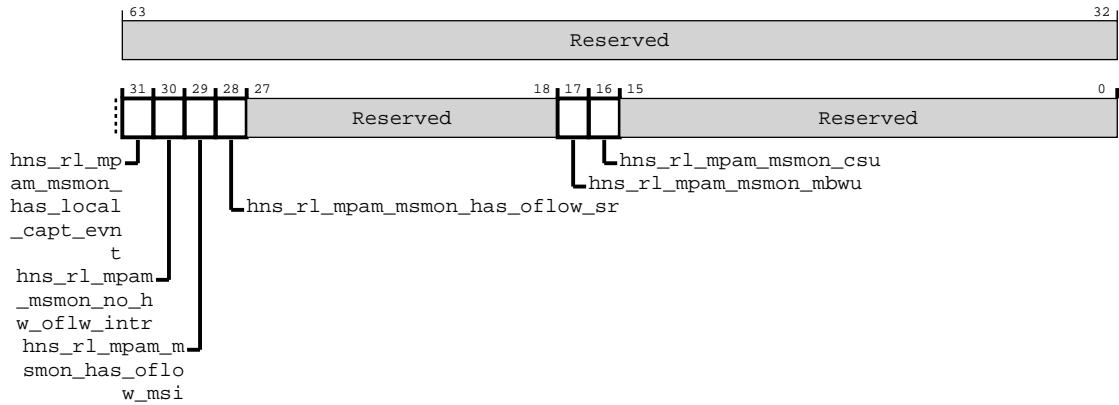


Table 5-125: cmn_hns_rl_mpam_msmon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rl_mpam_msmon_has_local_capt_evt</code>	Has the local capture event generator and the MSMON_CAPT_EVNT register.	RO	1'h1
[30]	<code>hns_rl_mpam_msmon_no_hw_oflw_intr</code>	0: HNF doesn't have hardwired MPAM overflow interrupt 1: HNF has have hardwired MPAM overflow interrupt	RO	1'b0
[29]	<code>hns_rl_mpam_msmon_has_oflow_msi</code>	0: HNF doesn't have support for MSI writes to signal MPAM monitor overflow interrupt 1: HNF has support for MSI writes to signal MPAM monitor overflow interrupt	RO	1'b0
[28]	<code>hns_rl_mpam_msmon_has_oflow_sr</code>	0: HNF doesn't have overflow status register 1: HNF has overflow status register	RO	1'b0
[27:18]	Reserved	Reserved	RO	-
[17]	<code>hns_rl_mpam_msmon_mbwu</code>	This component has a performance monitor for Memory Bandwidth Usage by PARTID and PMG.	RO	Configuration dependent
[16]	<code>hns_rl_mpam_msmon_csu</code>	This component has a performance monitor for Cache Storage Usage by PARTID and PMG.	RO	Configuration dependent
[15:0]	Reserved	Reserved	RO	-

5.2.3.44 cmn_hns_rl_mpam_csumon_idr

MPAM cache storage usage monitor ID register. This is banked separately.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8088

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-122: cmn_hns_rl_mpam_csumon_idr

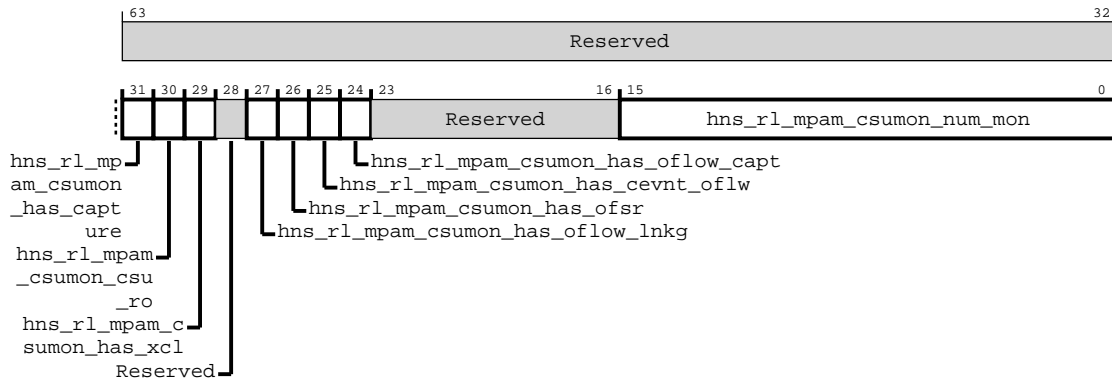


Table 5-126: cmn_hns_rl_mpam_csumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rl_mpam_csumon_has_capture</code>	0: MSMON_CSU_CAPTURE is not implemented and there is no support for capture events in this component's CSU monitor feature. 1: This component's CSU monitor feature has an MSMON_CSU_CAPTURE register for every MSMON_CSU and supports the capture event behaviour.	RO	1'h1
[30]	<code>hns_rl_mpam_csumon_csu_ro</code>	0: MSMON_CSU is read/write. 1: MSMON_CSU is read-only.	RO	1'b0
[29]	<code>hns_rl_mpam_csumon_has_xcl</code>	0: MSMON_CFG_CSU_FLT does not implement the XCL field 1: MSMON_CFG_CSU_FLT implements the XCL field	RO	1'b0
[28]	Reserved	Reserved	RO	-
[27]	<code>hns_rl_mpam_csumon_has_oflow_lnk</code>	0: HNF does not support CSU overflow linkage 1: HNF supports CSU overflow linkage	RO	1'b0
[26]	<code>hns_rl_mpam_csumon_has_ofsr</code>	0: MSMON_CSU_OFSR register is not implemented 1: MSMON_CSU_OFSR register is implemented	RO	1'b0
[25]	<code>hns_rl_mpam_csumon_has_cevnt_oflw</code>	0: HNF does not support MSMON_CFG_CSU_CTL.CEVNT_OFLW 1: HNF supports MSMON_CFG_CSU_CTL.CEVNT_OFLW	RO	1'b0

Bits	Name	Description	Type	Reset
[24]	hns_rl_mpam_csumon_has_oflow_capt	0: HNF does not support MSMON_CFG_CSU_CTL.OFLOW_CAPT 1: HNF supports MSMON_CFG_CSU_CTL.OFLOW_CAPT	RO	1'b0
[23:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_mpam_csumon_num_mon	The number of CSU monitoring counters implemented in this component.	RO	Configuration dependent

5.2.3.45 cmn_hns_rl_mpam_mbwumon_idr

MPAM memory bandwidth usage monitor ID register. This is banked separately.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8090

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-123: cmn_hns_rl_mpam_mbwumon_idr

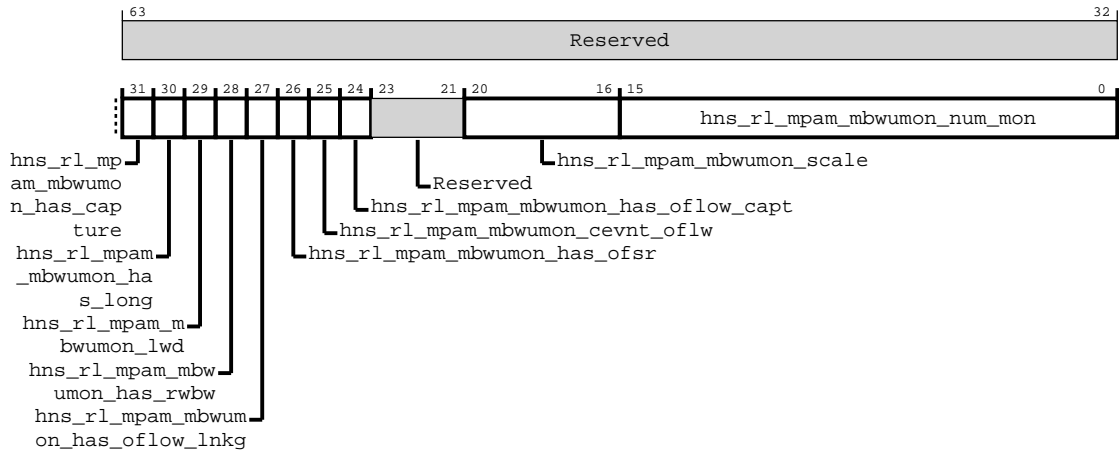


Table 5-127: cmn_hns_rl_mpam_mbwumon_idr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_mpam_mbwumon_has_capture	0: MSMON_MBWU_CAPTURE is not implemented and there is no support for capture events in this component's MBWU monitor feature. 1: This component's MBWU monitor feature has an MSMON_MBWU_CAPTURE register for every MSMON_MBWU and supports the capture event behavior.	RO	1'h0
[30]	hns_rl_mpam_mbwumon_has_long	0: MSMON_MBWU_L is not implemented. 1: MSMON_MBWU_L is implemented.	RO	1'b0
[29]	hns_rl_mpam_mbwumon_lwd	0: MSMON_MBWU_L has 44-bit VALUE field in bits [43:0]. 1: MSMON_MBWU_L has 63-bit VALUE field in bits [62:0].	RO	1'b0
[28]	hns_rl_mpam_mbwumon_has_rwbw	0: Read/write bandwidth selection is not implemented. 1: Read/write bandwidth selection is implemented.	RO	1'b0
[27]	hns_rl_mpam_mbwumon_has_oflow_lnkg	0: Does not support MSMON_CFG_MBWU_CTL.OFLOW_LNKG. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_LNKG.	RO	1'b0
[26]	hns_rl_mpam_mbwumon_has_ofsr	0: MSMON_MBWU_OFSR register is not implemented 1: MSMON_MBWU_OFSR register is implemented	RO	1'b0
[25]	hns_rl_mpam_mbwumon_cevnt_oflw	0: Does not support MSMON_CFG_MBWU_CTL.CEVNT_OFLW. 1: Support MSMON_CFG_MBWU_CTL.CEVNT_OFLW.	RO	1'b0
[24]	hns_rl_mpam_mbwumon_has_oflow_capt	0: Does not support MSMON_CFG_MBWU_CTL.OFLOW_CAPT. 1: Support MSMON_CFG_MBWU_CTL.OFLOW_CAPT.	RO	1'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	hns_rl_mpam_mbwumon_scale	Scaling of MSMON_MBWU.VALUE in bits.	RO	5'h0
[15:0]	hns_rl_mpam_mbwumon_num_mon	The number of MBWU monitoring counters implemented in this component.	RO	16'h0

5.2.3.46 cmn_hns_rl_mpam_ecr

MPAM Error Control Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h80F0

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-124: cmn_hns_rl_mpam_ecr

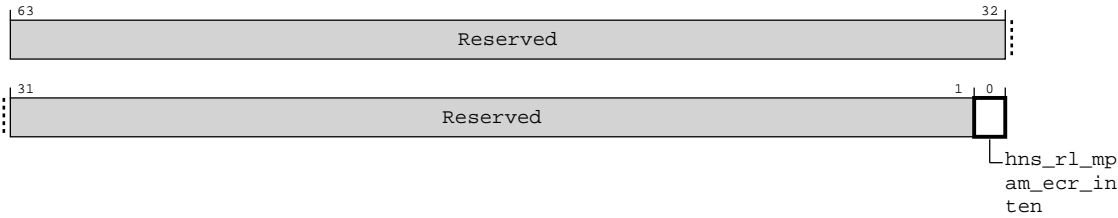


Table 5-128: cmn_hns_rl_mpam_ecr attributes

Bits	Name	Description	Type	Reset
[63:1]	Reserved	Reserved	RO	-
[0]	hns_rl_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

5.2.3.47 cmn_hns_rl_mpam_esr

MPAM Error Status Register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h80F8

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-125: cmn_hns_rl_mpam_esr

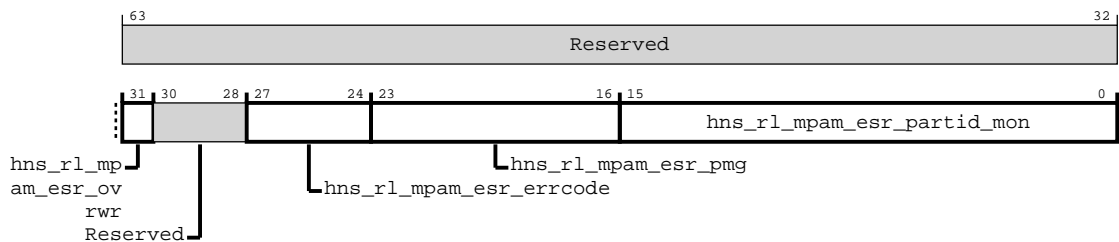


Table 5-129: cmn_hns_rl_mpam_esr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rl_mpam_esr_ovrwr</code>	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	-

Bits	Name	Description	Type	Reset
[27:24]	hns_rl_mpam_esr_errcode	Error code	RW	4'h0
[23:16]	hns_rl_mpam_esr_pmg	PMG captured if the error code captures PMG, otherwise 0x0000.	RW	8'h0
[15:0]	hns_rl_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

5.2.3.48 cmn_hns_rl_mpamcfg_part_sel

MPAM partition configuration selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8100

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-126: cmn_hns_rl_mpamcfg_part_sel

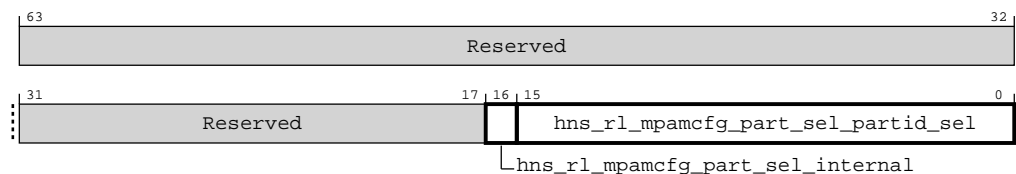


Table 5-130: cmn_hns_rl_mpamcfg_part_sel attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[16]	hns_rl_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]	hns_rl_mpamcfg_part_sel_partid_sel	Selects the partition ID to configure.	RW	16'h0

5.2.3.49 cmn_hns_rl_mpamcfg_cmax

MPAM cache maximum capacity partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8108

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-127: cmn_hns_rl_mpamcfg_cmax

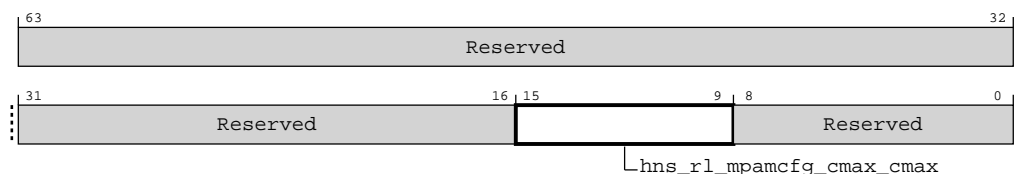


Table 5-131: cmn_hns_rl_mpamcfg_cmax attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:9]	hns_rl_mpamcfg_cmax_cmax	Maximum cache capacity usage in fixed-point fraction of the cache capacity by the partition selected by MPAMCFG_PART_SEL.	RW	7'b1111111
[8:0]	Reserved	Reserved	RO	-

5.2.3.50 cmn_hns_rl_mpamcfg_mbw_min

MPAM memory minimum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8200

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-128: cmn_hns_rl_mpamcfg_mbw_min

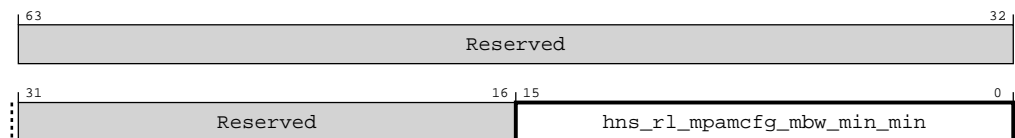


Table 5-132: cmn_hns_rl_mpamcfg_mbw_min attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_mpamcfg_mbw_min_min	Memory minimum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.51 cmn_hns_rl_mpamcfg_mbw_max

MPAM memory maximum bandwidth partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8208

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-129: cmn_hns_rl_mpamcfg_mbw_max

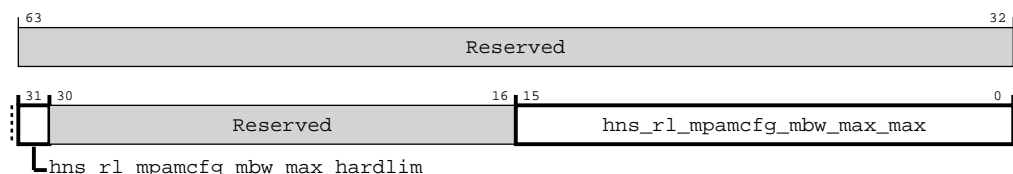


Table 5-133: cmn_hns_rl_mpamcfg_mbw_max attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rl_mpamcfg_mbw_max_hardlim</code>	0: When MAX bandwidth is exceeded, the partition may contend with a low preference for downstream bandwidth beyond its maximum bandwidth. 1: When MAX bandwidth is exceeded, the partition may not be use any more bandwidth until its memory bandwidth measurement falls below the maximum limit.	RW	1'h0
[30:16]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[15:0]	hns_rl_mpamcfg_mbw_max_max	Memory maximum bandwidth allocated to the partition selected by MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.52 cmn_hns_rl_mpamcfg_mbw_winwd

MPAM memory bandwidth partitioning window width register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8220

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-130: cmn_hns_rl_mpamcfg_mbw_winwd

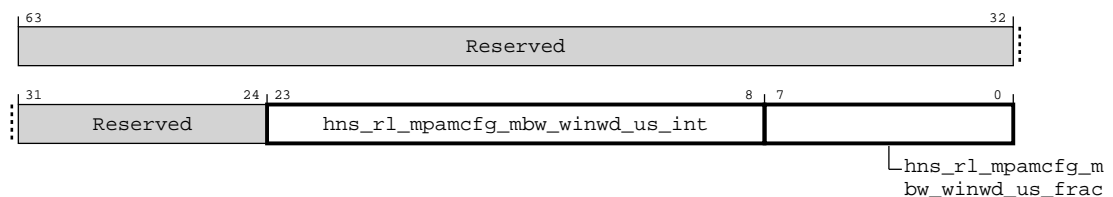


Table 5-134: cmn_hns_rl_mpamcfg_mbw_winwd attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:8]	hns_rl_mpamcfg_mbw_winwd_us_int	Memory bandwidth accounting period integer microseconds.	RW	16'h0
[7:0]	hns_rl_mpamcfg_mbw_winwd_us_frac	Memory bandwidth accounting period fractions of a microsecond.	RW	8'h0

5.2.3.53 cmn_hns_rl_mpamcfg_pri

MPAM priority partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8400

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-131: cmn_hns_rl_mpamcfg_pri

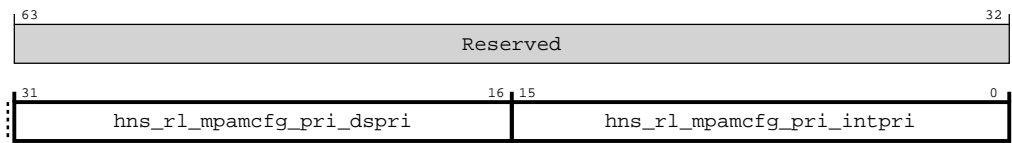


Table 5-135: cmn_hns_rl_mpamcfg_pri attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	hns_rl_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	16'h0
[15:0]	hns_rl_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	16'h0

5.2.3.54 cmn_hns_rl_mpamcfg_mbw_prop

Memory bandwidth proportional stride partitioning configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8500

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-132: cmn_hns_rl_mpamcfg_mbw_prop

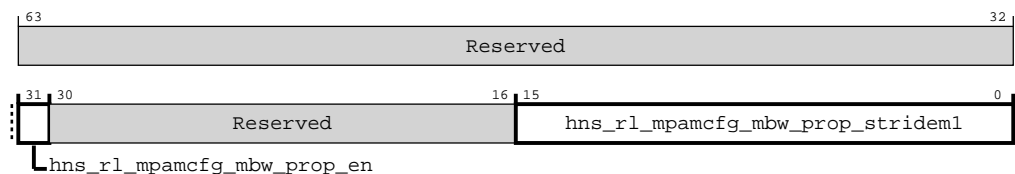


Table 5-136: cmn_hns_rl_mpamcfg_mbw_prop attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rl_mpamcfg_mbw_prop_en</code>	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 <code>STRIDEM1</code> .	RW	1'h0
[30:16]	Reserved	Reserved	RO	-
[15:0]	<code>hns_rl_mpamcfg_mbw_prop_stridem1</code>	Normalized cost of a bandwidth consumption by the partition. <code>STRIDEM1</code> is the stride for the partition minus one.	RW	16'h0

5.2.3.55 cmn_hns_rl_mpamcfg_intpartid

MPAM internal partition narrowing configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8600

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-133: cmn_hns_rl_mpamcfg_intpartid

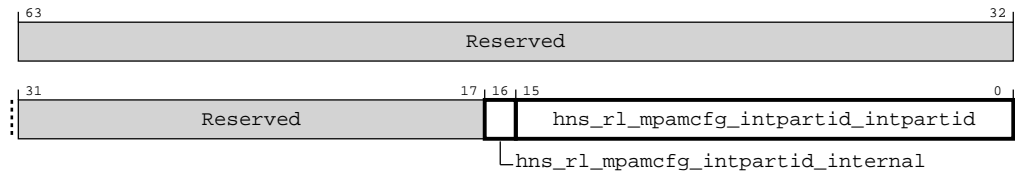


Table 5-137: cmn_hns_rl_mpamcfg_intpartid attributes

Bits	Name	Description	Type	Reset
[63:17]	Reserved	Reserved	RO	-
[16]	<code>hns_rl_mpamcfg_intpartid_internal</code>	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]	<code>hns_rl_mpamcfg_intpartid_intpartid</code>	This field contains the intPARTID mapped to the reqPARTID in MPAMCFG_PART_SEL.	RW	16'h0

5.2.3.56 cmn_hns_rl_msmon_cfg_mon_sel

Memory system performance monitor selection register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8800

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-134: cmn_hns_rl_msmon_cfg_mon_sel

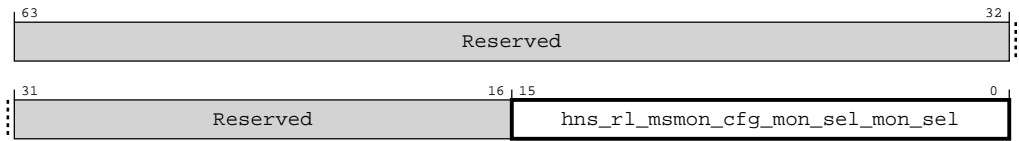


Table 5-138: cmn_hns_rl_msmon_cfg_mon_sel attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_msmon_cfg_mon_sel_mon_sel	Selects the performance monitor to configure.	RW	16'h0

5.2.3.57 cmn_hns_rl_msmon_capt_evnt

Memory system performance monitoring capture event generation register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8808

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-135: cmn_hns_rl_msmon_capt_evnt

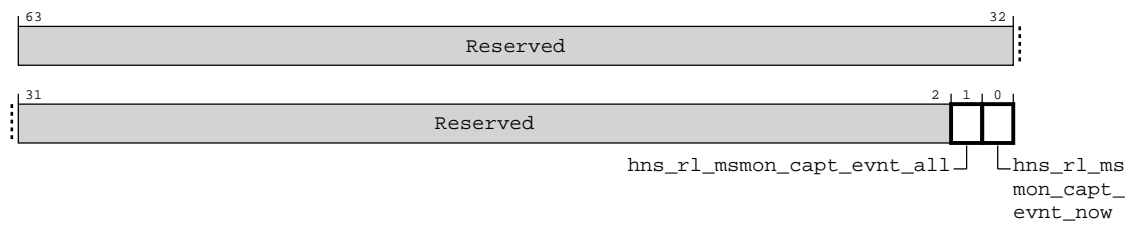


Table 5-139: cmn_hns_rl_msmon_capt_evnt attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1]	hns_rl_msmon_capt_evnt_all	In secure version, if ALL written as 1 and NOW is also written as 1, signal a capture event to secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to secure monitors in this memory system component with CAPT_EVNT = 7. In non-secure version if NOW is written as 1, signal a capture event to non-secure monitors in this memory system component with CAPT_EVNT = 7. In root version, if ALL written as 1 and NOW is also written as 1, signal a capture event to root, realm, secure and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to root monitors in this memory system component with CAPT_EVNT = 7. In realm version, if ALL written as 1 and NOW is also written as 1, signal a capture event to realm and non-secure monitors in this memory system component with CAPT_EVNT = 7. If written as 0 and NOW is written as 1, signal a capture event to realm monitors in this memory system component with CAPT_EVNT = 7.	RW	1'h0
[0]	hns_rl_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

5.2.3.58 cmn_hns_rl_msmon_cfg_csuflt

Memory system performance monitor configure cache storage usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8810

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-136: cmn_hns_rl_msmon_cfg_csuflt

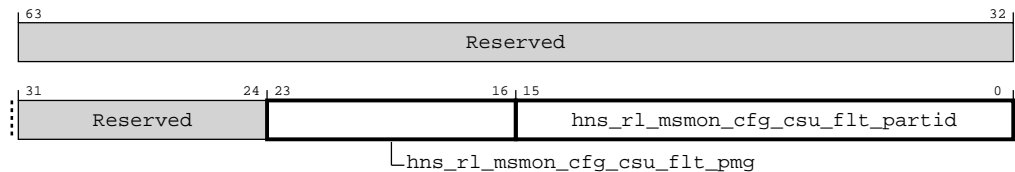


Table 5-140: cmn_hns_rl_msmon_cfg_csuflt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	hns_rl_msmon_cfg_csuflt_pmg	Configures the cache storage usage performance monitor to a PMG. The monitor selected by MSMON_CFG_MON_SEL.MON_SEL counts or measures storage usage by cache lines labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	hns_rl_msmon_cfg_csuflt_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

5.2.3.59 cmn_hns_rl_msmon_cfg_csuctl

Memory system performance monitor configure cache storage usage monitor control register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8818

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-137: cmn_hns_rl_msmon_cfg_csu_ctl

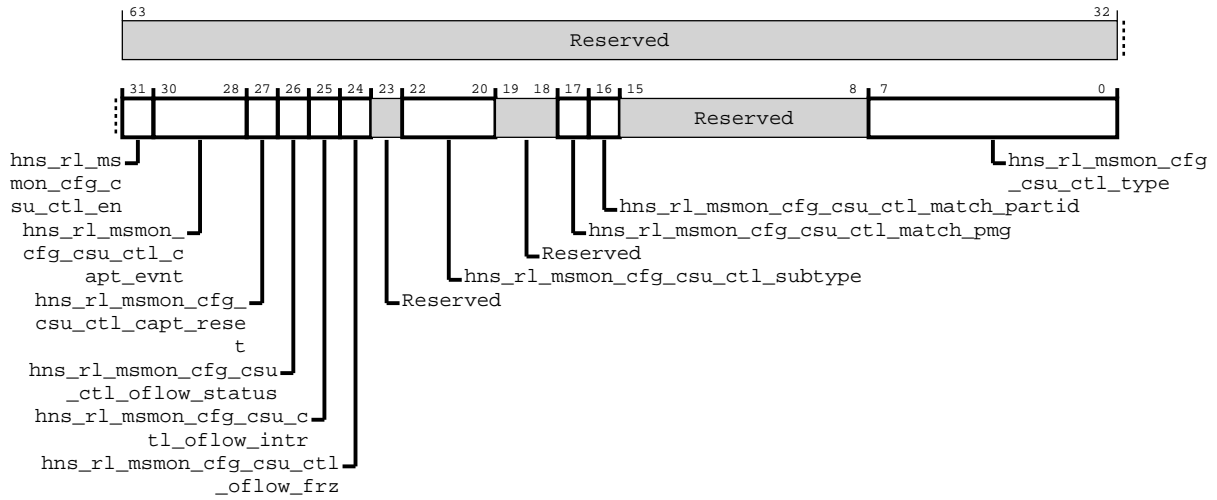


Table 5-141: cmn_hns_rl_msmon_cfg_csu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	<code>hns_rl_msmon_cfg_csu_ctl_en</code>	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]	<code>hns_rl_msmon_cfg_csu_ctl_capt_evt</code>	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]	<code>hns_rl_msmon_cfg_csu_ctl_capt_reset</code>	Capture is not implemented for the CSU monitor type.	RW	1'h0
[26]	<code>hns_rl_msmon_cfg_csu_ctl_oflow_status</code>	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	<code>hns_rl_msmon_cfg_csu_ctl_oflow_intr</code>	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	<code>hns_rl_msmon_cfg_csu_ctl_oflow_frz</code>	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]	Reserved	Reserved	RO	-
[22:20]	<code>hns_rl_msmon_cfg_csu_ctl_subtype</code>	Not currently used for CSU monitors, but reserved for future use.	RW	3'h0
[19:18]	Reserved	Reserved	RO	-
[17]	<code>hns_rl_msmon_cfg_csu_ctl_match_pmg</code>	0: Monitor storage used by all PMG values. 1: Only monitor storage used with the PMG value matching <code>MSMON_CFG_CSU_FLT.PMG</code> .	RW	1'h0
[16]	<code>hns_rl_msmon_cfg_csu_ctl_match_partid</code>	0: Monitor storage used by all PARTIDs. 1: Only monitor storage used with the PARTID matching <code>MSMON_CFG_CSU_FLT.PARTID</code> .	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	<code>hns_rl_msmon_cfg_csu_ctl_type</code>	Read-only: Constant type indicating the type of the monitor. CSU monitor is <code>TYPE = 0x43</code> .	RW	8'h43

5.2.3.60 cmn_hns_rl_msmon_cfg_mbwuflt

Memory system performance monitor configure memory bandwidth usage monitor filter register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8820

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-138: cmn_hns_rl_msmon_cfg_mbwuflt

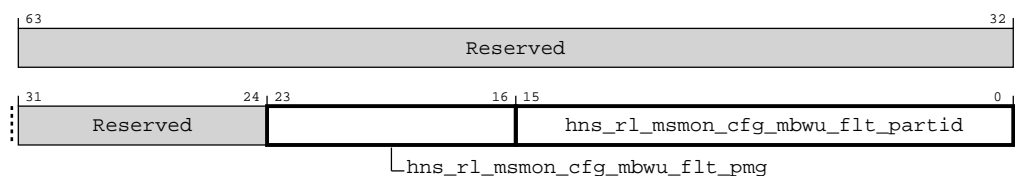


Table 5-142: cmn_hns_rl_msmon_cfg_mbwuflt attributes

Bits	Name	Description	Type	Reset
[63:24]	Reserved	Reserved	RO	-
[23:16]	<code>hns_rl_msmon_cfg_mbwuflt_pmg</code>	Configures the memory bandwidth usage performance monitor to a PMG. The monitor selected by <code>MSMON_CFG_MON_SEL.MON_SEL</code> counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	8'h0
[15:0]	<code>hns_rl_msmon_cfg_mbwuflt_partid</code>	Configures the memory bandwidth usage performance monitor to a PARTID. The monitor selected by <code>MSMON_CFG_MON_SEL.MON_SEL</code> counts or measures the memory bandwidth used by requests labelled with both the configured PARTID and PMG.	RW	16'h0

5.2.3.61 cmn_hns_rl_msmon_cfg_mbwu_ctl

Memory system performance monitor configure memory bandwidth usage monitor control register.
This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8828

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-139: cmn_hns_rl_msmon_cfg_mbwu_ctl

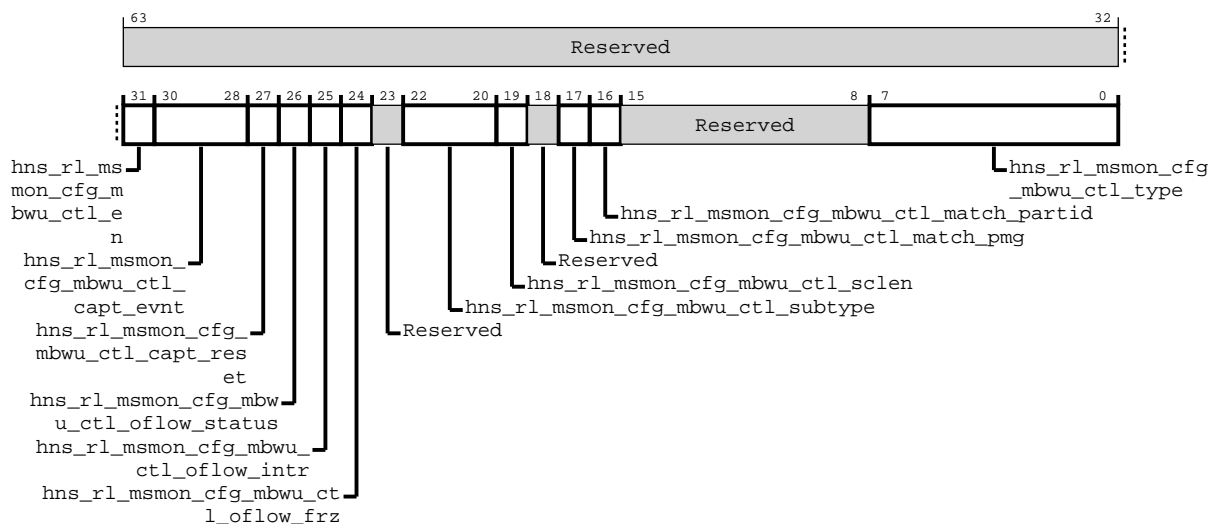


Table 5-143: cmn_hns_rl_msmon_cfg_mbwu_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_msmon_cfg_mbwu_ctl_en	0: The monitor is disabled and must not collect any information. 1: The monitor is enabled to collect information according to its configuration.	RW	1'h0
[30:28]	hns_rl_msmon_cfg_mbwu_ctl_capt_evt	Select the event that triggers capture from the following: 0: No capture event is triggered. 1: External capture event 1 (optional but recommended)	RW	3'h0
[27]	hns_rl_msmon_cfg_mbwu_ctl_capt_reset	0: Monitor is not reset on capture. 1: Monitor is reset on capture.	RW	1'h0
[26]	hns_rl_msmon_cfg_mbwu_ctl_oflow_status	0: No overflow has occurred. 1: At least one overflow has occurred since this bit was last written.	RW	1'h0
[25]	hns_rl_msmon_cfg_mbwu_ctl_oflow_intr	0: No interrupt. 1: On overflow, an implementation-specific interrupt is signalled.	RW	1'h0
[24]	hns_rl_msmon_cfg_mbwu_ctl_oflow_frz	0: Monitor count wraps on overflow. 1: Monitor count freezes on overflow. The frozen value may be 0 or another value if the monitor overflowed with an increment larger than 1.	RW	1'h0
[23]	Reserved	Reserved	RO	-
[22:20]	hns_rl_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	3'h0
[19]	hns_rl_msmon_cfg_mbwu_ctl_sclen	0: MSMON_MBWU.VALUE has bytes counted by the monitor instance. 1: MSMON_MBWU.VALUE has bytes counted by the monitor instance, shifted right by MPAMF_MBWUMON_IDR.SCALE.	RW	1'h0
[18]	Reserved	Reserved	RO	-
[17]	hns_rl_msmon_cfg_mbwu_ctl_match_pmg	0: Monitor bandwidth used by all PMG values. 1: Only monitor bandwidth used with the PMG value matching MSMON_CFG_CSU_FLT.PMG.	RW	1'h0
[16]	hns_rl_msmon_cfg_mbwu_ctl_match_partid	0: Monitor bandwidth used by all PARTIDs. 1: Only monitor bandwidth used with the PARTID matching MSMON_CFG_MBWU_FLT.PARTID.	RW	1'h0
[15:8]	Reserved	Reserved	RO	-
[7:0]	hns_rl_msmon_cfg_mbwu_ctl_type	Read-only: Constant type indicating the type of the monitor. MBWU monitor is TYPE = 0x42.	RW	8'h42

5.2.3.62 cmn_hns_rl_msmon_csu

Memory system performance monitor cache storage usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8840

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-140: cmn_hns_rl_msmon_csu

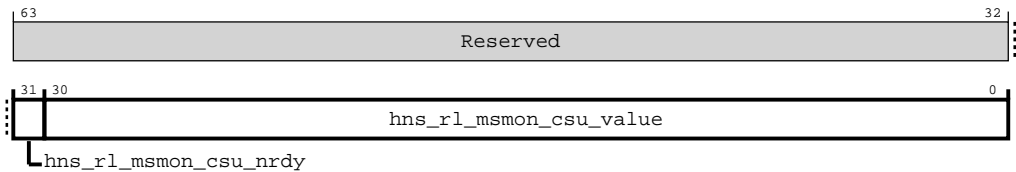


Table 5-144: cmn_hns_rl_msmon_csu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_msmon_csu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rl_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

5.2.3.63 cmn_hns_rl_msmon_csu_capture

Memory system performance monitor cache storage usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8848

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-141: cmn_hns_rl_msmon_csu_capture

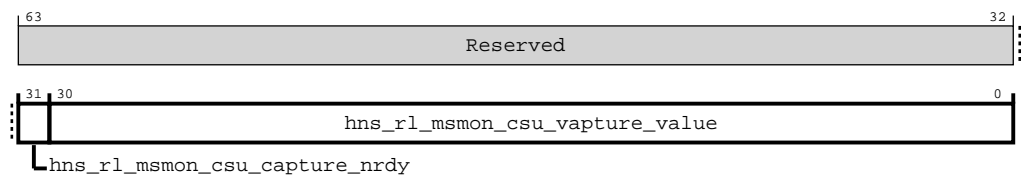


Table 5-145: cmn_hns_rl_msmon_csu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_msmon_csu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rl_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

5.2.3.64 cmn_hns_rl_msmon_mbwu

Memory system performance monitor memory bandwidth usage monitor register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8860

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-142: cmn_hns_rl_msmon_mbwu

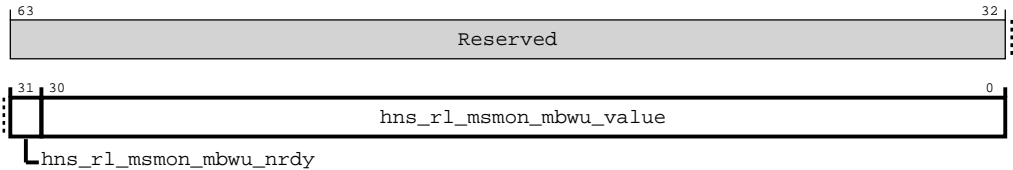


Table 5-146: cmn_hns_rl_msmon_mbwu attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_msmon_mbwu_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rl_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

5.2.3.65 cmn_hns_rl_msmon_mbwu_capture

Memory system performance monitor memory bandwidth usage capture register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h8868

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-143: cmn_hns_rl_msmon_mbwu_capture

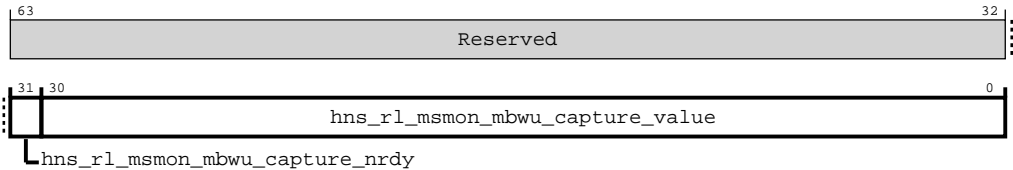


Table 5-147: cmn_hns_rl_msmon_mbwu_capture attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	hns_rl_msmon_mbwu_capture_nrdy	Indicates that the monitor does not have accurate data, possibly because insufficient time has elapsed since the monitor was last configured.	RW	1'h0
[30:0]	hns_rl_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

5.2.3.66 cmn_hns_rl_mpamcfg_cpbm

MPAM cache portion bitmap partition configuration register. This register is banked separately

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h9000

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Realm space and is accessible using Realm and Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-144: cmn_hns_rl_mpamcfg_cpbm

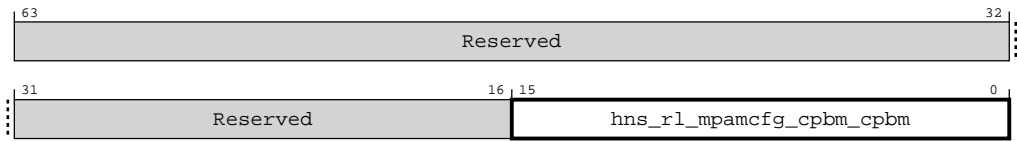


Table 5-148: cmn_hns_rl_mpamcfg_cpbm attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	hns_rl_mpamcfg_cpbm_cpbm	Bitmap of portions of cache capacity allocable by the partition selected by MPAMCFG_PART_SEL. NOTE: CPBM can not be all zeros for any PARTID.	RW	16'hFFFF

5.2.4 HN-S register descriptions

This section lists the HN-S registers.

5.2.4.1 cmn_hns_node_info

Provides component identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-145: cmn_hns_node_info

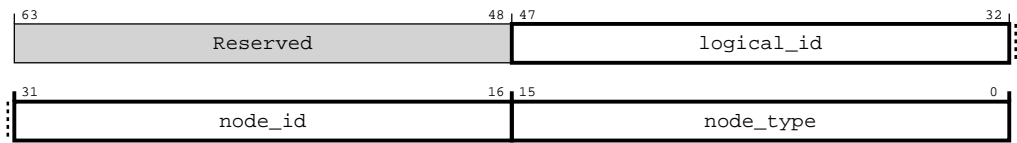


Table 5-149: cmn_hns_node_info attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47:32]	logical_id	\$logical_id_description	RO	Configuration dependent
[31:16]	node_id	\$node_id_description	RO	Configuration dependent
[15:0]	node_type	\$node_type_description	RO	Configuration dependent

5.2.4.2 cmn_hns_child_info

Provides component child identification information.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h80

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-146: cmn_hns_child_info

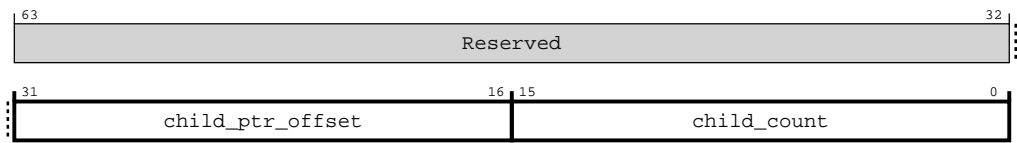


Table 5-150: cmn_hns_child_info attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:16]	child_ptr_offset	Starting register offset which contains pointers to the child nodes	RO	16'h0
[15:0]	child_count	Number of child nodes; used in discovery process	RO	16'b0

5.2.4.3 cmn_hns_scr

Secure register access override.

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

This register is owned in the Secure space and is accessible using Secure or Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-147: cmn_hns_scr

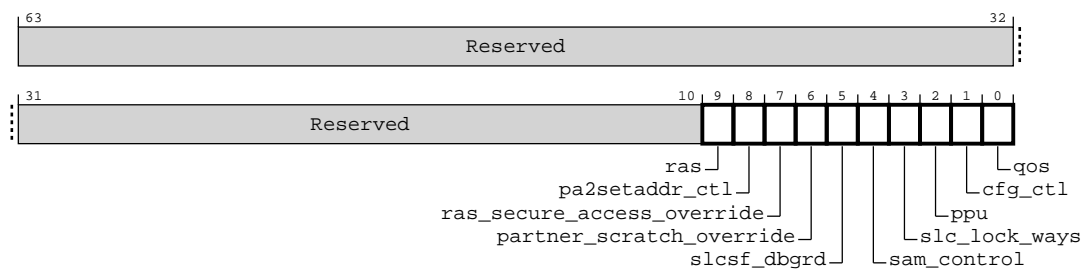


Table 5-151: cmn_hns_scr attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9]	ras	Allow Secure override of the RAS registers	RW	1'b0
[8]	pa2setaddr_ctl	Allow Secure override of the PA2SETADDR registers	RW	1'b0
[7]	ras_secure_access_override	Allow Secure override of the RAS ERRMISC<n> registers	RW	1'b0
[6]	partner_scratch_override	Allows Secure override of the partner scratch registers	RW	1'b0
[5]	slcsf_dbgdr	Allows Secure override of the SLC/SF debug read registers	RW	1'b0
[4]	sam_control	Allows Secure override of the HN-F SAM control registers	RW	1'b0
[3]	slc_lock_ways	Allows Secure override of the cache way locking registers	RW	1'b0
[2]	ppu	Allows Secure override of the power policy registers	RW	1'b0
[1]	cfg_ctl	Allows Secure override of the configuration control register (cmn_hns_cfg_ctl)	RW	1'b0
[0]	qos	Allows Secure override of the QoS registers	RW	1'b0

5.2.4.4 cmn_hns_rcr

Root register access override.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h988

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-148: cmn_hns_rcr

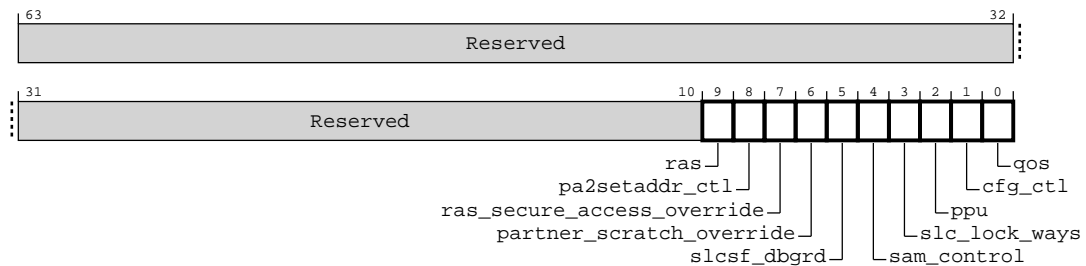


Table 5-152: cmn_hns_rcr attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9]	ras	Allow Root override of the RAS registers	RW	1'b0
[8]	pa2setaddr_ctl	Allows Root override of the PA2SETADDR registers	RW	1'b0
[7]	ras_secure_access_override	Allows Root override of the RAS ERRMISC<n> registers	RW	1'b0
[6]	partner_scratch_override	Allows Root override of the Partner scratch registers	RW	1'b0
[5]	slcsf_dbgrd	Allows Root override of the SLC/SF debug read registers	RW	1'b0
[4]	sam_control	Allows Root override of the HN-F SAM control registers	RW	1'b0
[3]	slc_lock_ways	Allows Root override of the cache way locking registers	RW	1'b0
[2]	ppu	Allows Root override of the power policy registers	RW	1'b0
[1]	cfg_ctl	Allows Root override of the configuration control register (cmn_hns_cfg_ctl)	RW	1'b0
[0]	qos	Allows Root override of the QoS registers	RW	1'b0

5.2.4.5 cmn_hns_unit_info

Provides component identification information for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h900

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-149: cmn_hns_unit_info

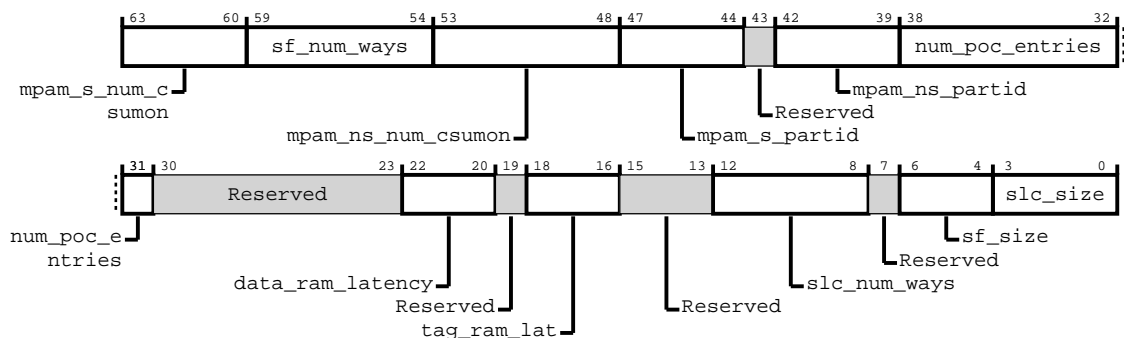


Table 5-153: cmn_hns_unit_info attributes

Bits	Name	Description	Type	Reset
[63:60]	mpam_s_num_csumon	Number of Secure Cache Storage Usage Monitors for MPAM	RO	Configuration dependent
[59:54]	sf_num_ways	Number of cache ways in the SF	RO	-
[53:48]	mpam_ns_num_csumon	Number of Non-Secure Cache Storage Usage Monitors for MPAM	RO	Configuration dependent

Bits	Name	Description	Type	Reset
[47:44]	mpam_s_partid	MPAM Secure supported PARTIDs 4'b0000: 1 S PARTID 4'b0001: Reserved 4'b0010: Reserved 4'b0011: 8 S PARTID 4'b0100: 16 S PARTID	RO	-
[43]	Reserved	Reserved	RO	-
[42:39]	mpam_ns_partid	MPAM Non-Secure supported PARTIDs 4'b0000: 1 NS PARTID 4'b0001: Reserved 4'b0010: Reserved 4'b0011: Reserved 4'b0100: Reserved 4'b0101: 32 NS PARTID 4'b0110: 64 NS PARTID 4'b0111: 128 NS PARTID 4'b1000: 256 NS PARTID 4'b1001: 512 NS PARTID	RO	-
[38:31]	num_poc_entries	Number of POCQ entries	RO	Configuration dependent
[30:23]	Reserved	Reserved	RO	-
[22:20]	data_ram_latency	SLC data RAM latency (in cycles)	RO	-
[19]	Reserved	Reserved	RO	-
[18:16]	tag_ram_lat	SLC tag RAM latency (in cycles)	RO	-
[15:13]	Reserved	Reserved	RO	-
[12:8]	slc_num_ways	Number of cache ways in the SLC	RO	-
[7]	Reserved	Reserved	RO	-
[6:4]	sf_size	SF size 3'b000: (32KB * sf_num_ways) 3'b001: (64KB * sf_num_ways) 3'b010: (128KB * sf_num_ways) 3'b011: (256KB * sf_num_ways) 3'b100: (512KB * sf_num_ways)	RO	-
[3:0]	slc_size	SLC size 4'b0000: No SLC 4'b0001: 128KB 4'b0010: 256KB 4'b0011: 512KB 4'b0100: 1MB 4'b0101: 1.5MB 4'b0110: 2MB 4'b0111: 3MB 4'b1000: 4MB 4'b1001: 384KB	RO	-

5.2.4.6 cmn_hns_unit_info_1

Provides component identification information for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h908

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-150: cmn_hns_unit_info_1

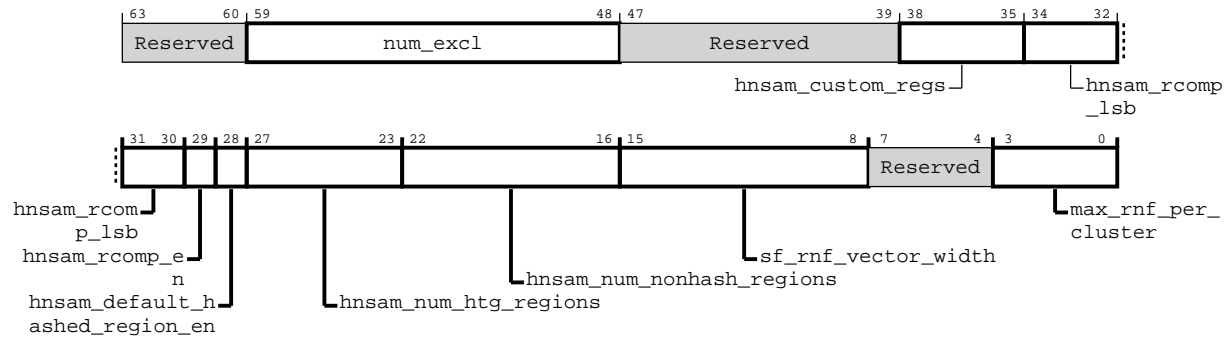


Table 5-154: cmn_hns_unit_info_1 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59:48]	num_excl	Number of exclusive monitors	RO	-
[47:39]	Reserved	Reserved	RO	-
[38:35]	hnsam_custom_regs	Number of customer specific registers for customer implemented logic	RO	Configuration dependent
[34:30]	hnsam_rcomp_lsb	Defines the minimum size of HTG when POR_HNSAM_RCOMP_EN_PARAM = 1, 20 value defines minimum size as 1MB and 26 value defines minimum size as 64MB	RO	Configuration dependent
[29]	hnsam_rcomp_en	Enable Range based address comparison for HNSAM HTG/Nonhashed groups. Program start address and end address	RO	Configuration dependent
[28]	hnsam_default_hashed_region_en	Enable default hashed group for HNSAM. To support backward compatible, set this parameter	RO	Configuration dependent
[27:23]	hnsam_num_htg_regions	Number of HTG regions supported by the HNSAM	RO	Configuration dependent
[22:16]	hnsam_num_nonhash_regions	Number of non-hashed regions supported by the HNSAM	RO	Configuration dependent
[15:8]	sf_rnf_vector_width	Total Number of bits in RNF tracking vector in the Snoop Filter (Total SF_VEC_WIDTH = (TOTAL_RNF/HNS_MAX_CLUSTER_PARAM)+HNS_SF_ADD_VECTOR_WIDTH)	RO	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3:0]	max_rnf_per_cluster	Describes the maximum number of RN-F's in a single cluster	RO	Configuration dependent

5.2.4.7 cmn_hns_cfg_ctl

Functions as the configuration control register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA00

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-151: cmn_hns_cfg_ctl

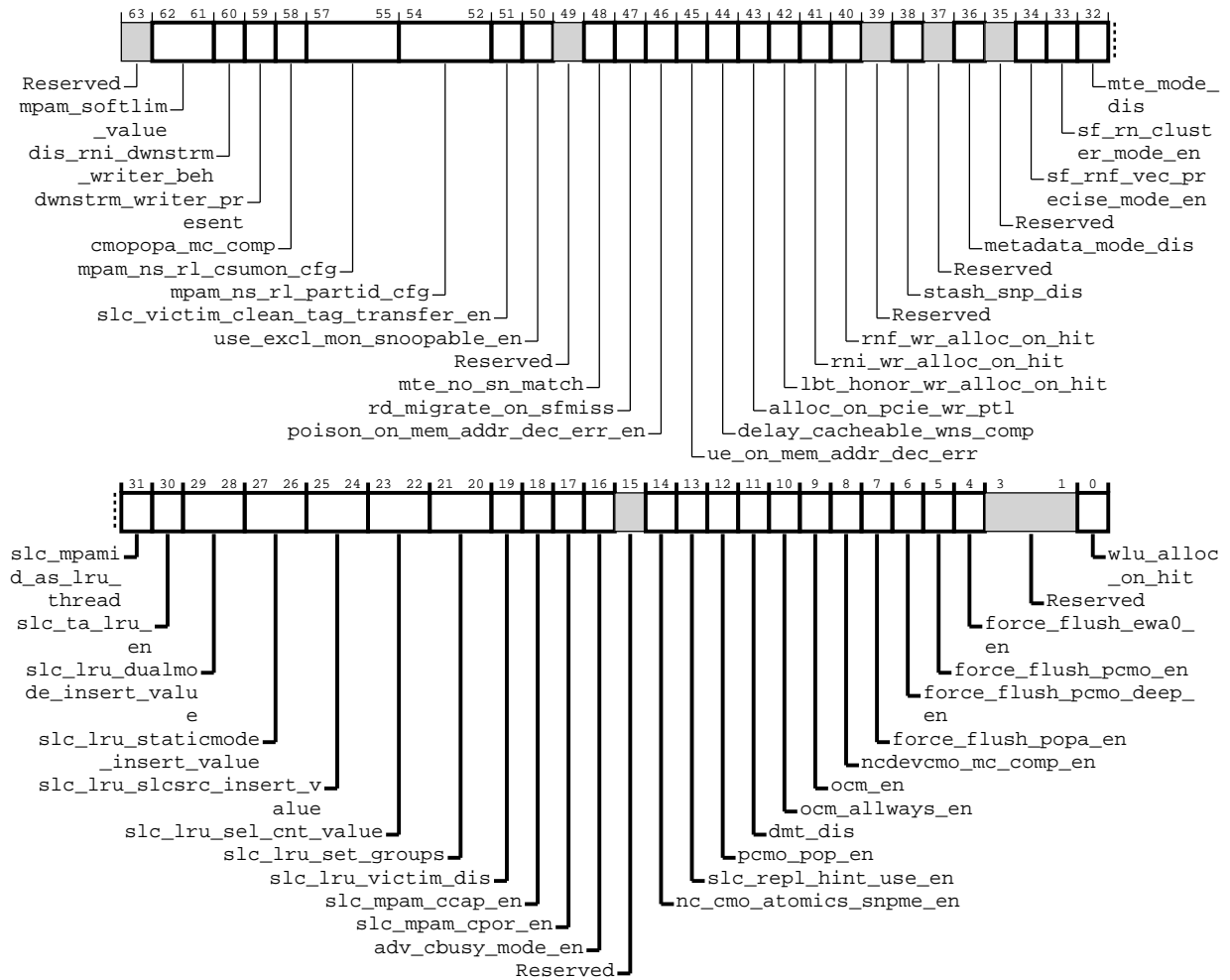


Table 5-155: cmn_hns_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:61]	<code>mpam_softlim_value</code>	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: Default is 3.13% below hardlimit. If CMAX value set is at or below 12.5%, soft limit is ignored.	RW	2'b01
[60]	<code>dis_rni_dwnstrm_writer_beh</code>	When set, HNS doesn't do SN read for NC requests from RNI/RND when clean data is available in SLC or upstream RNs Only applicable when <code>dwnstrm_writer_present</code> is set	RW	1'b0
[59]	<code>dwnstrm_writer_present</code>	When set, HNS does SN read for NC requests when clean data is available in SLC or upstream RNs	RW	1'b0
[58]	<code>cmopopa_mc_comp</code>	When set, HN-F sends completion for CMOPoPA after completion from SN	RW	1'b0

Bits	Name	Description	Type	Reset
[57:55]	mpam_ns_rl_csumon_cfg	MPAM Non-Secure/Realm CSUMON configuration based on a combined total of (HNS_MPAM_NS_NUM_CSUMON_PARAM +HNS_MPAM_RL_NUM_CSUMON_PARAM) CSUMONs 3'b000: 1 CSUMON allocated to Realm; Remaining CSUMONs allocated to Non-Secure 3'b001: 1 CSUMON allocated to Non-Secure; Remaining CSUMONs allocated to Realm 3'b010: 1/2 CSUMONs allocated to Non-Secure; 1/2 to Realm. Remaining 1 CSUMON allocated to Non-Secure 3'b011: 3/4 CSUMONs allocated to Non-Secure; 1/4 to Realm. Remaining 1 CSUMON allocated to Non-Secure 3'b100: 1/4 CSUMONs allocated to Non-Secure; 3/4 to Realm. Remaining 1 CSUMON allocated to Non-Secure	RW	3'b0
[54:52]	mpam_ns_rl_partid_cfg	MPAM Non-Secure/Realm PARTID configuration based on a combined total of (HNS_MPAM_NS_PARTID_MAX_PARAM +HNS_MPAM_RL_PARTID_MAX_PARAM) PARTIDs 3'b000: 1 PARTID allocated to Realm; Remaining PARTIDs allocated to Non-Secure 3'b001: 1 PARTID allocated to Non-Secure; Remaining PARTIDs allocated to Realm 3'b010: 1/2 PARTIDs allocated to Non-Secure; 1/2 to Realm. Remaining 1 PARTID allocated to Non-Secure 3'b011: 3/4 PARTIDs allocated to Non-Secure; 1/4 to Realm. Remaining 1 PARTID allocated to Non-Secure 3'b100: 1/4 PARTIDs allocated to Non-Secure; 3/4 to Realm. Remaining 1 PARTID allocated to Non-Secure	RW	3'b0
[51]	slc_victim_clean_tag_transfer_en	When set, HNS propagates clean tag to SN when SLC victim has clean tag	RW	1'b0
[50]	use_excl_mon_snoopable_en	When set, HNS uses exclusive monitor for snoopable traffic in imprecise SF modes	RW	1'b0
[49]	Reserved	Reserved	RO	-
[48]	mte_no_sn_match	When set, HNS does MTE match locally without propagating to SN	RW	1'b0
[47]	rd_migrate_on_sfmiss	Migrates a read from LCC/SLC if sf miss	RW	1'b1
[46]	poison_on_mem_addr_dec_err_en	When set, set poison in read data for CXL address decode error	RW	1'b1
[45]	ue_on_mem_addr_dec_err	Log CXL address decode error as UE in error register	RW	1'b0
[44]	delay_cacheable_wns_comp	Sends late completion for cacheable WriteNoSnoop	RW	1'b0
[43]	alloc_on_pcie_wr_ptl	Forces HBT PCIE partial writes to allocate in SLC	RW	1'b0
[42]	lbt_honor_wr_alloc_on_hit	Forces LBT Write requests to honor wlu_alloc_on_hit, rnf_wr_alloc_on_hit, and rni_wr_alloc_on_hit	RW	1'b0
[41]	rni_wr_alloc_on_hit	Forces RNI Write requests to allocate if the line hit in SLC	RW	1'b0
[40]	rnf_wr_alloc_on_hit	Forces RNF Write requests to allocate if the line hit in SLC	RW	1'b0
[39]	Reserved	Reserved	RO	-
[38]	stash_snp_dis	Disables stashing snoop in HNS when set to 1'b1	RW	1'b0
[37]	Reserved	Reserved	RO	-
[36]	metadata_mode_dis	Disables the METADATA features in HNS when set to 1'b1	RW	1'b0
[35]	Reserved	Reserved	RO	-
[34]	sf_rnf_vec_precise_mode_en	Enables the snoop filter's precise RNF vector in clustered mode when set to 1'b1	RW	1'b1
[33]	sf_rn_cluster_mode_en	Enables the snoop filter clustering of the RN-F ID's using programmable registers	RW	1'b1
[32]	mte_mode_dis	Disables the MTE features in HNS when set to 1'b1	RW	1'b0
[31]	slc_mpamid_as_lru_thread	Use MPAM PARTID as ThreadID for Thread Aware 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

Bits	Name	Description	Type	Reset
[30]	slc_ta_lru_en	Thread Aware eLRU enable 1'b0: ThreadID used for eLRU is zero. 1'b1: ThreadID used for eLRU is based on MPAMID or LPID+LID. Note: If SLC size is less than 256KB, this bit is ignore.	RW	1'b0
[29:28]	slc_lru_dualmode_insert_value	Insertion value for Dual mode eLRU NOTE: Default is 2'b11.	RW	2'b11
[27:26]	slc_lru_staticmode_insert_value	Insertion value for Static mode eLRU NOTE: Default is 2'b10.	RW	2'b10
[25:24]	slc_lru_slcsrc_insert_value	Insertion value if SLC source bit is set NOTE: Default is 2'b00.	RW	2'b00
[23:22]	slc_lru_sel_cnt_value	Selection counter value for eLRU to determine which group policy is more effective 2'b00: Sel counter is like an 8-bit range; upper limit is 255; middle point is 128 2'b01: Sel counter is like a 9-bit range; upper limit is 511; middle point is 256 2'b10: Sel counter is like a 10-bit range; upper limit is 1023; middle point is 512 2'b11: Sel counter is like an 11-bit range; upper limit is 2047; middle point is 1024 NOTE: Default is 10-bit with counter reset to a value of 512.	RW	2'b10
[21:20]	slc_lru_set_groups	Number of sets in monitor group for enhance LRU 2'b00: 16 2'b01: 32 2'b10: 64 2'b11: 128 NOTE: Default is 32 sets per monitor group. If cache size is small (128KB or less), there would be only one set per group.	RW	2'b01
[19]	slc_lru_victim_dis	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
[18]	slc_mpam_ccap_en	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]	slc_mpam_cpor_en	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]	adv_cbusy_mode_en	Enables the advanced features of HNS CBusy handling	RW	1'b0
[15]	Reserved	Reserved	RO	-
[14]	nc_cmo_atomics_snpme_en	When set to 1, all incoming non-cachable atomics and cmo's from RNF will be back-snooped	RW	1'b0
[13]	slc_repl_hint_use_en	1'b0: Interconnect generated SLC Replacement hints are used for eLRU. 1'b1: RN-F provided SLC Replacement hints are used for eLRU.	RW	1'b1
[12]	pcmo_pop_en	Terminates PCMO in HNS when this bit is set to 1'b1	RW	1'b0
[11]	dmt_dis	Disables DMT when set	RW	1'b0
[10]	ocm_allways_en	Enables all SLC ways with OCM	RW	1'b0
[9]	ocm_en	Enables region locking with OCM support	RW	1'b0
[8]	ncdevcmo_mc_comp_en	Disables HN-F completion when set NOTE: When set, HN-F sends completion for the following transactions received after completion from SN: 1. Non-cacheable WriteNoSnp 2. Device WriteNoSnp 3. CMO (cache maintenance operations) CONSTRAINT: When this bit is set, por_rni_cfg_ctl.dis_ncwr_stream and por_rnd_cfg_ctl.dis_ncwr_stream must also be set.	RW	1'b0
[7]	force_flush_popa_en	Generate PoPA request for SLC and SF flush generated SN writes. CONSTRAINT: force_flush_popa_en is valid only if ABF CleanInvalid mode is set.	RW	1'b0
[6]	force_flush_pcmo_deep_en	Make PCMO request for SLC and SF flush generated SN writes as Deep PCMO. CONSTRAINT: hns_force_flush_pcmo_deep_en is valid only if hns_force_flush_pcmo_en bit is set. CONSTRAINT: This bit can be set only if ALL SNs in the system support deep attribute.	RW	1'b0
[5]	force_flush_pcmo_en	Generate PCMO request for SLC and SF flush generated SN writes	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	force_flush_ewa0_en	Force SLC and SF flush to use EWA 0 for SN writes	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	wlu_alloc_on_hit	Forces WLU requests to allocate if the line hit in SLC	RW	1'b0

5.2.4.8 cmn_hns_aux_ctl

Functions as the auxiliary control register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA08

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-152: cmn_hns_aux_ctl

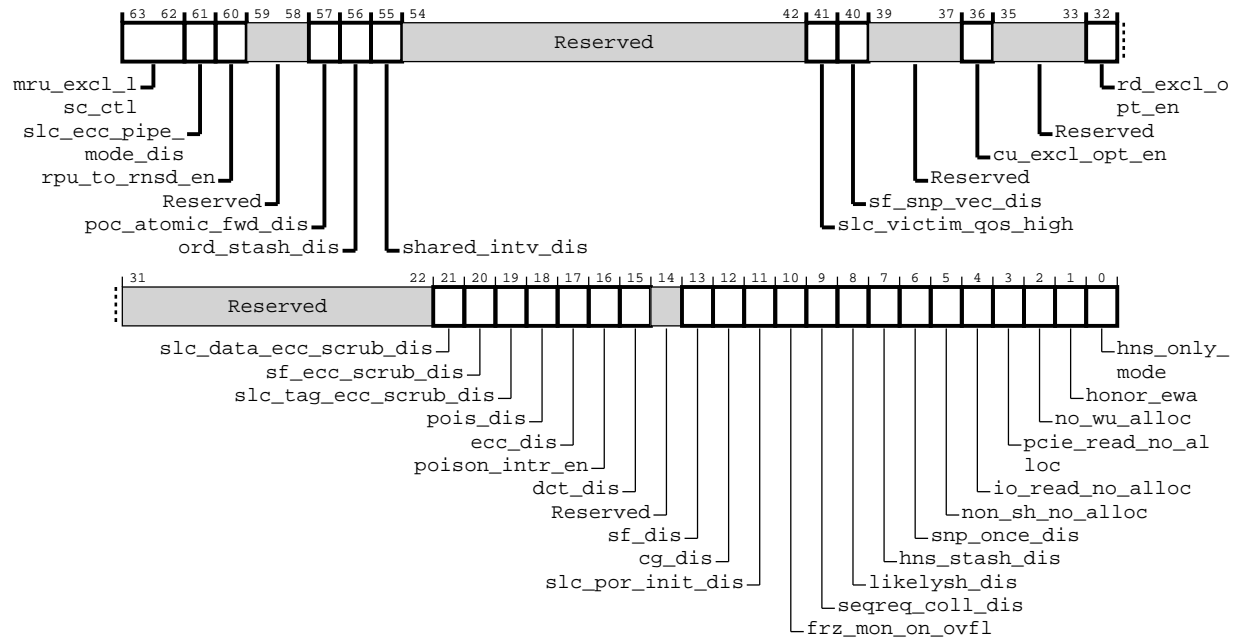


Table 5-156: cmn_hns_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:62]	mru_excl_lsc_ctl	MRU Exclusive control for LSC.	RW	2'b00
[61]	slc_ecc_pipe_mode_dis	Disables inline ECC pipe mode in SLC. CONSTRAINT: Must be programmed at boot time.	RW	1'b1
[60]	rpu_to_rnsd_en	Enables HN-F to treat ReadPrefUnique ops as ReadNotSharedDirty	RW	1'b0
[59:58]	Reserved	Reserved	RO	-
[57]	poc_atomic_fwd_dis	Disable the atomic data forwarding in POCQ	RW	1'b0
[56]	ord_stash_dis	Disables stash operation for ordered write stash requests	RW	1'b0
[55]	shared_intv_dis	Disables snoop requests to CHIB RN-F with shared copy	RW	Configuration dependent
[54:42]	Reserved	Reserved	RO	-
[41]	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]	sf_snp_vec_dis	Disables SF snoop vector when set	RW	1'b0
[39:37]	Reserved	Reserved	RO	-
[36]	cu_excl_opt_en	CleanUnique exclusive optimization enable	RW	1'b1
[35:33]	Reserved	Reserved	RO	-
[32]	rd_excl_opt_en	ReadNotSharedDirty exclusive optimization enable	RW	1'b0
[31:22]	Reserved	Reserved	RO	-
[21]	slc_data_ecc_scrub_dis	Disables data single-bit ECC error scrubbing for non-migrating reads when set	RW	1'b1
[20]	sf_ecc_scrub_dis	Disables SF tag single-bit ECC error scrubbing when set	RW	1'b0
[19]	slc_tag_ecc_scrub_dis	Disables SLC tag single-bit ECC error scrubbing when set	RW	1'b0

Bits	Name	Description	Type	Reset
[18]	pois_dis	Disables parity error data poison when set	RW	1'b0
[17]	ecc_dis	Disables SLC and SF ECC generation/detection when set	RW	1'b0
[16]	poison_intr_en	Enables reporting an interrupt by HN-F when poison is detected at SLC	RW	Configuration dependent
[15]	dct_dis	Disables DCT when set	RW	Configuration dependent
[14]	Reserved	Reserved	RO	-
[13]	sf_dis	Disables SF	RW	1'b0
[12]	cg_dis	Disables HN-F architectural clock gates	RW	1'b0
[11]	slc_por_init_dis	Disables SLC and SF initialization on Reset	RW	1'b0
[10]	frz_mon_on_ovfl	Freezes the exclusive monitors	RW	1'b0
[9]	seqreq_coll_dis	-	RW	1'b0
[8]	likelysh_dis	Disables Likely Shared based allocations	RW	1'b0
[7]	hns_stash_dis	Disables HN-F stash support	RW	Configuration dependent
[6]	snp_once_dis	When set, disables SnpOnce and converts to SnpShared	RW	Configuration dependent
[5]	non_sh_no_alloc	Disables SLC allocation for non-shareable cacheable transactions when set	RW	1'b0
[4]	io_read_no_alloc	When set, disables ReadOnce and ReadNoSnp allocation in SLC from RN-Is	RW	1'b0
[3]	pcie_read_no_alloc	When set, disables ReadOnce and ReadNoSnp allocation in SLC from PCIE	RW	1'b0
[2]	no_wu_alloc	Disables WriteUnique/WriteLineUnique allocations in SLC when set	RW	1'b0
[1]	honor_ewa	When set, postpones completion for writes where EWA=0 in the request until HN-F receives completion from MC or SBSX	RW	1'b1
[0]	hns_only_mode	Enables HN-F only mode; disables SLC and SF when set	RW	1'b0

5.2.4.9 cmn_hns_aux_ctl_1

Functions as the auxiliary control register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA10

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-153: cmn_hns_aux_ctl_1

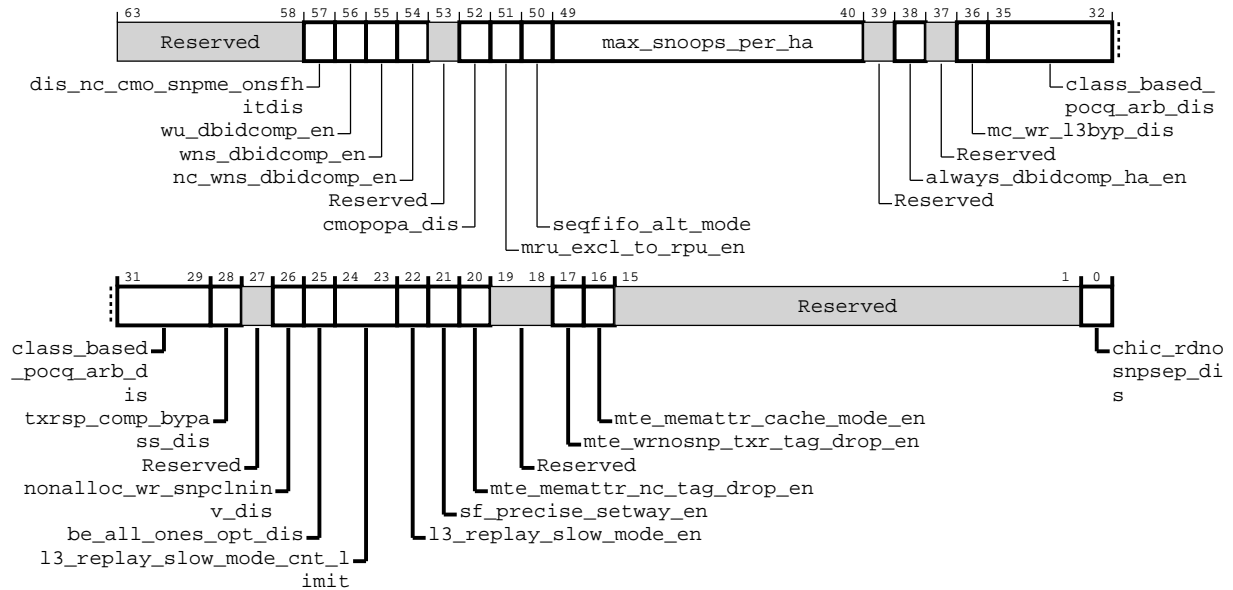


Table 5-157: cmn_hns_aux_ctl_1 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57]	dis_nc_cmo_snpme_onsfh	When set to 1, disables nc cmo snpme functionality to self-snoop on SF hit or SF disable	RW	1'b0
[56]	itdis	When set, HNS will combine DBID and Comp response for all WriteUnique requests to post SLC/SF lookup and snoops. Only applicable to HBT transactions	RW	1'b0
[55]	wu_dbidcomp_en	When set, HNS will combine DBID and Comp response for all WriteNoSnoop requests to post SLC/SF lookup and snoops	RW	1'b0
[54]	wns_dbidcomp_en	When set, HNS will combine DBID and Comp response for all NC WriteNoSnoop requests to post SLC/SF lookup and snoops	RW	1'b0
[53]	nc_wns_dbidcomp_en	When set, HNS will combine DBID and Comp response for all NC WriteNoSnoop requests to post SLC/SF lookup and snoops	RW	1'b0
[52]	Reserved	Reserved	RO	-
[51]	cmopopa_dis	When set, HN-F downgrades CleanInvalidPoPA to CleanInvalid	RW	1'b0
[50]	mru_excl_to_rpu_en	When set, Excl MRU will send RPU flow when fail and SF hit EU but not self hit	RW	1'b0
[49]	seqfifo_alt_mode	HNS will use alternate mode to select SEQ entries.	RW	1'b0
[40:49]	max_snoops_per_ha	HNS will use the register value instead of the parameter to control Max snoops per HA	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[39]	Reserved	Reserved	RO	-
[38]	always_dbidcomp_ha_en	When set, HNS will combine DBID and Comp response for all writeunique and writenosnoop requests from CXHA to post SLC/SF lookup and snoops	RW	1'b0
[37]	Reserved	Reserved	RO	-
[36]	mc_wr_l3byp_dis	When set, disables l3 bypass path to mc request for writes	RW	1'b0
[35:29]	class_based_pocq_arb_dis	Disables Class based arbitration for various POCQ arbiters. For each bit: 1'b0: Use Class based arbitration. 1'b1: Use QoS based arbitration. Legacy mode. [35]: POCQ entry selection for SN Static Credit Grant return. [34]: POCQ entry selection for SLC/SF pipeline request. [33]: POCQ entry selection for TXRSP. [32]: POCQ entry selection for TXDAT. [31]: POCQ entry selection for TXREQ. [30]: POCQ entry selection for ADQ. [29]: Reserved for future use.	RW	7'b0000000
[28]	txrsp_comp_bypass_dis	When set, TXRSP COMP bypass gets disabled for WEOE/EVICT	RW	1'b1
[27]	Reserved	Reserved	RO	-
[26]	nonalloc_wr_snpclninv_dis	Disable the snp type of snp_cln_inv on non-allocating writes. Send snp_uniq instead	RW	1'b0
[25]	be_all_ones_opt_dis	Disable the optimizations related to BE=1's hint on WR_PTL from RNI	RW	1'b0
[24:23]	l3_replay_slow_mode_cnt_limit	L3 arbitration throttle count limit, when enabled. 00: L3 Throttle is enabled after 512 setway haz replays 01: L3 Throttle is enabled after 1024 setway haz replays 10: L3 Throttle is enabled after 2048 setway haz replays 11: L3 Throttle is enabled after 4096 setway haz replays	RW	2'b00
[22]	l3_replay_slow_mode_en	Enables L3 arbitration slow mode in case of constant replays, when set to 1'b1	RW	1'b0
[21]	sf_precise_setway_en	Enables Precise setway hazard, when set to 1'b1	RW	1'b0
[20]	mte_memattr_nc_tag_drop_en	Enables HNS to drop any dirty tags for Non-Cacheable memory, when set to 1'b1	RW	1'b0
[19:18]	Reserved	Reserved	RO	-
[17]	mte_wrnosnp_trx_tag_drop_en	When set to 1'b1, HNS will drop clean tags from a WriteNoSnp with tagop Transfer	RW	1'b0
[16]	mte_memattr_cache_mode_en	When set to 1'b1, it enables HNS to convert Non-cacheable requests to cacheable if MTE tags are required	RW	1'b1
[15:1]	Reserved	Reserved	RO	-
[0]	chic_rdnosnpsep_dis	Disables separation of Data and Comp in CHIC mode	RW	1'b0

5.2.4.10 cmn_hns_cbusy_limit_ctl

Cbusy threshold limits for POCQ entries.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA18

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-154: cmn_hns_cbusy_limit_ctl

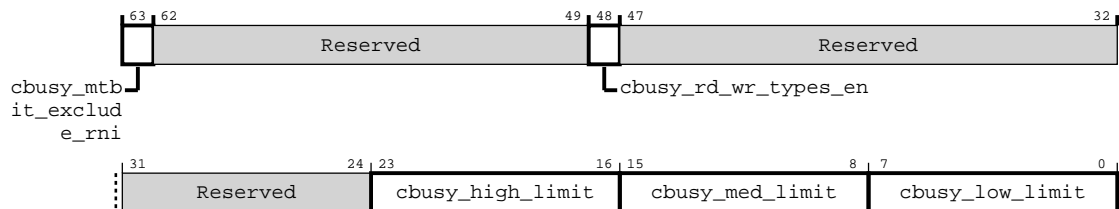


Table 5-158: cmn_hns_cbusy_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63]	<code>cbusy_mtbit_exclude_rni</code>	Exclude RNI sources in multi-source mode	RW	1'b0
[62:49]	Reserved	Reserved	RO	-
[48]	<code>cbusy_rd_wr_types_en</code>	When set, CBusy for Reads and Writes are handled independently. The thresholds specified in this register are used for Read request types in POCQ	RW	1'b0
[47:24]	Reserved	Reserved	RO	-
[23:16]	<code>cbusy_high_limit</code>	POCQ limit for CBusy High	RW	Configuration dependent
[15:8]	<code>cbusy_med_limit</code>	POCQ limit for CBusy Med	RW	Configuration dependent
[7:0]	<code>cbusy_low_limit</code>	POCQ limit for CBusy Low	RW	Configuration dependent

5.2.4.11 cmn_hns_txrsp_arb_weight_ctl

TXRSP arbitration weight controls.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA20

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-155: cmn_hns_txrsp_arb_weight_ctl

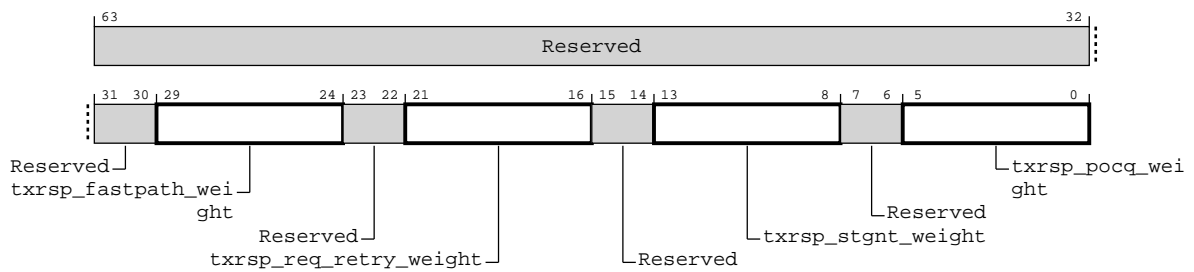


Table 5-159: cmn_hns_txrsp_arb_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29:24]	<code>txrsp_fastpath_weight</code>	Fastpath response weights for TXRSP channel	RW	6'b111111
[23:22]	Reserved	Reserved	RO	-
[21:16]	<code>txrsp_req_retry_weight</code>	Request retry response weights for TXRSP channel	RW	6'b000001
[15:14]	Reserved	Reserved	RO	-
[13:8]	<code>txrsp_stgnt_weight</code>	Static Credit Grant response weights for TXRSP channel	RW	6'b000001
[7:6]	Reserved	Reserved	RO	-
[5:0]	<code>txrsp_pocq_weight</code>	POCQ response weights for TXRSP channel	RW	6'b000001

5.2.4.12 cmn_hns_cbusy_mode_ctl

Control register for additional CBusy controls

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA28

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-156: cmn_hns_cbusy_mode_ctl

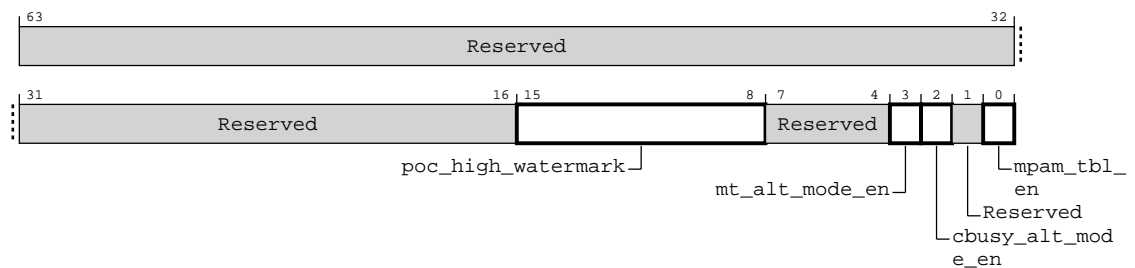


Table 5-160: cmn_hns_cbusy_mode_ctl attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	poc_high_watermark	Number of POCQ entries when it is considered high occupancy	RW	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3]	mt_alt_mode_en	Enable CBusy[2] alternate reporting mode: 1'b0: POCQ has requests from more than one source 1'b1: POCQ Occupancy is higher than the poc_high_watermark	RW	1'b0

Bits	Name	Description	Type	Reset
[2]	cbusy_alt_mode_en	Enables an alternate mode of SN CBusy[1:0] capture for mpam_tbl_en=1 mode: 1'b0: For each MPAM partID, CBusy[1:0] = SN_CBusy[1:0] 1'b1: For each MPAM partID, CBusy[1] = SN's CBusy[2], CBusy[0] = (SN_CBusy[1] & SN_CBusy[0])	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	mpam_tbl_en	Enables cbusy reporting based on MPAM part ID	RW	1'b0

5.2.4.13 cmn_hns_lbt_cfg_ctl

Functions as the configuration control register for HN-F. Only applicable to LBT transactions

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hA30

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-157: cmn_hns_lbt_cfg_ctl

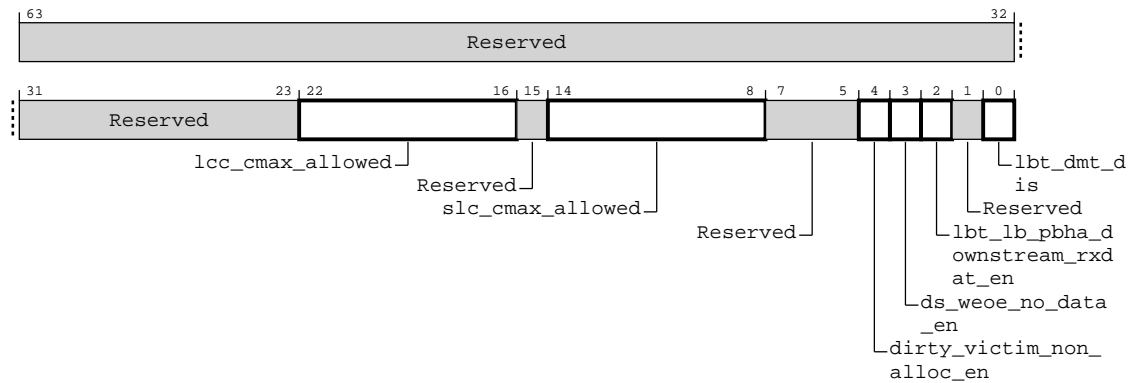


Table 5-161: cmn_hns_lbt_cfg_ctl attributes

Bits	Name	Description	Type	Reset
[63:23]	Reserved	Reserved	RO	-
[22:16]	lcc_cmax_allowed	Maximum cache capacity usage in fixed-point fraction of the cache capacity by LBT lines	RW	7'b1111111
[15]	Reserved	Reserved	RO	-
[14:8]	slc_cmax_allowed	Maximum cache capacity usage in fixed-point fraction of the cache capacity by HBT lines	RW	7'b1111111
[7:5]	Reserved	Reserved	RO	-
[4]	dirty_victim_non_alloc_en	When HNS issues dirty CopyBack writes for LCC victim or SFBI, set non-allocating type	RW	1'b0
[3]	ds_weoe_no_data_en	When HNS issues WriteEvictOrEvict downstream, force no data transfer if config bit is set	RW	1'b0
[2]	lbt_lb_pbha_downstream_rxd_at_en	Takes LB/PBHA values from downstream RXDAT for LBT lines	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	lbt_dmt_dis	Disables DMT when set	RW	1'b0

5.2.4.14 cmn_hns_lbt_aux_ctl

Functions as the auxiliary control register for HN-F. Only applicable to LBT transactions

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hA38

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-158: cmn_hns_lbt_aux_ctl

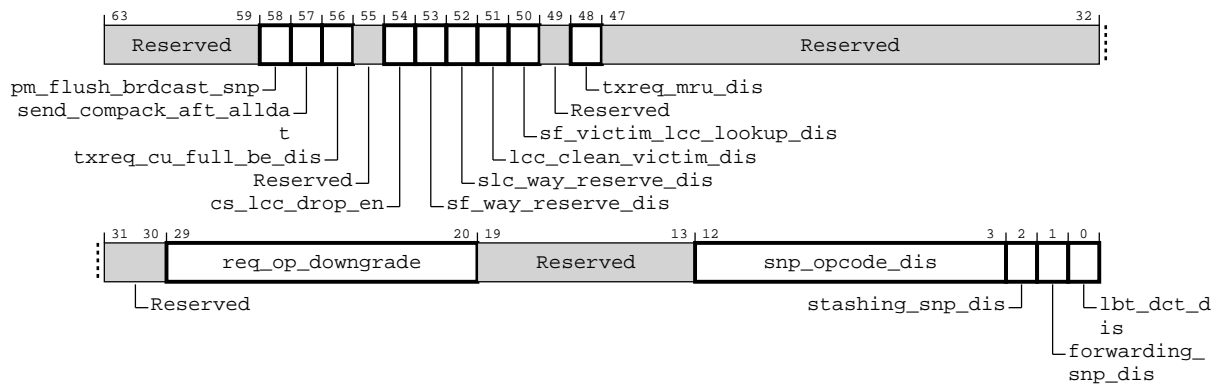


Table 5-162: cmn_hns_lbt_aux_ctl attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58]	pm_flush_brdcast_snp	Enables broadcast snoop to all local RNFs when PM flushes LBT SF entry. Can program to 1'b0 only when it's guaranteed that no coherence traffic comes to HNS during PM flush.	RW	1'b1
[57]	send_compack_aft_allda	Enables sending CompAck after all data beats are received	RW	1'b0
[56]	txreq_cu_full_be_dis	Disables LCC always getting ownership by CleanUnique for streaming writes with full BE.	RW	1'b0
[55]	Reserved	Reserved	RO	-
[54]	cs_lcc_drop_en	Enables LCC to drop clean copy after writing dirty data for CleanShared.	RW	1'b1
[53]	sf_way_reserve_dis	Disables SF reserved way for HBT lines in HNS mode and allow LBT lines to take all ways in SF.	RW	1'b0
[52]	slc_way_reserve_dis	Disables SLC reserved way for HBT lines in HNS mode and allow LBT lines to take all ways in system cache.	RW	1'b0
[51]	lcc_clean_victim_dis	Disables LCC sending clean eviction to Home Node.	RW	1'b0
[50]	sf_victim_lcc_lookup_dis	Disables LCC lookup for a SF victim.	RW	1'b0
[49]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[48]	txreq_mru_dis	Disables sending MRU opcode downstream. Use RDUNIQ instead	RW	1'b0
[47:30]	Reserved	Reserved	RO	-
[29:20]	req_op_downgrade	Downgrades req opcode when set [20]: Change READSHARED to READUNIQ [21]: Change READNOTSHARED DIRTY to READUNIQ [22]: Change READPREFERUNIQ to READUNIQ [23]: Change READONCEMKNV to READONCE [24]: Change READONCECLNINV to READONCE [25]: Change MAKEUNIQ to CLNUNIQ [26]: Change MAKEREADUNIQ to READUNIQ [27]: Change WRITEEVICTOREVICT to EVICT [28]: Change MAKEINVALID to CLEANINVALID [29]: Change WRITEUNIQUEFULL to non-allocating WRITEUNIQUEPTL	RW	10'h000
[19:13]	Reserved	Reserved	RO	-
[12:3]	snp_opcode_dis	Disables support for RXSNP different snoop opcodes and changes to similar snoop opcode when set [3]: Change SNPSTASHSHARED to SNPQUERY [4]: Change SNPSTASHUNIQUE to SNPQUERY [5]: Change SNPMAKEINVALIDSTASH to SNPMAKEINVALID [6]: Change SNPCLEANSHARED to SNPCLEANINVALID [7]: Change SNPPREFERUNIQUE(FWD) to SNPUNIQUE [8]: Change SNPNOTSHARED DIRTY(FWD) to SNPUNIQUE [9]: Change SNPCLEAN(FWD) to SNPUNIQUE [10]: Change SNPUNIQUESTASH to SNPUNIQUE [11]: Change SNPONCE(FWD) to SNPUNIQUE [12]: Change SNPSHARED(FWD) to SNPUNIQUE	RW	10'h000
[2]	stashing_snp_dis	Disables Stashing type of snoops when set for RXSNP	RW	1'b1
[1]	forwarding_snp_dis	Disables Forwarding type of snoops when set for RXSNP	RW	1'b1
[0]	lbt_dct_dis	Disables DCT when set	RW	1'b0

5.2.4.15 cmn_hns_ppu_pwpr

Functions as the power policy register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h1900

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ppu

Secure group override

cmn_hns_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ppu bit and cmn_hns_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-159: cmn_hns_ppu_pwpr

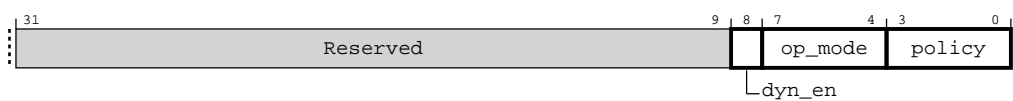


Table 5-163: cmn_hns_ppu_pwpr attributes

Bits	Name	Description	Type	Reset
[31:9]	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

5.2.4.16 cmn_hns_ppu_pwsr

Provides power status information for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h1908

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-160: cmn_hns_ppu_pwsr

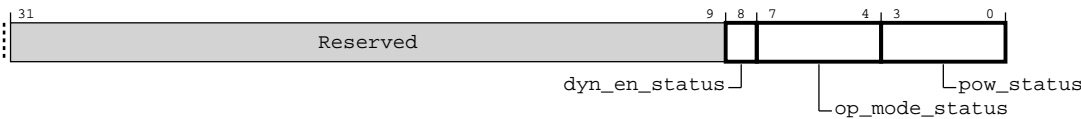


Table 5-164: cmn_hns_ppu_pwsr attributes

Bits	Name	Description	Type	Reset
[31:9]	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

5.2.4.17 cmn_hns_ppu_misr

Functions as the power miscellaneous input current status register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h1914

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-161: cmn_hns_ppu_misr

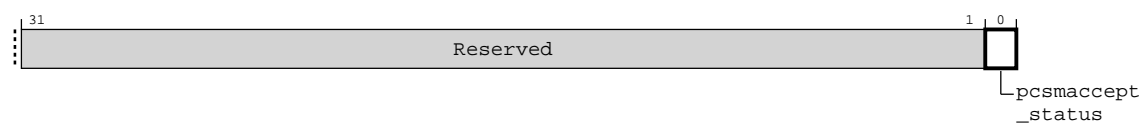


Table 5-165: cmn_hns_ppu_misr attributes

Bits	Name	Description	Type	Reset
[31:1]	Reserved	Reserved	RO	-
[0]	pcsmaccept_status	HN-F RAM PCSMACCEPT status	RO	1'b0

5.2.4.18 cmn_hns_ppu_idr0

Provides identification information for the HN-F PPU.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h28B0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-162: cmn_hns_ppu_idr0

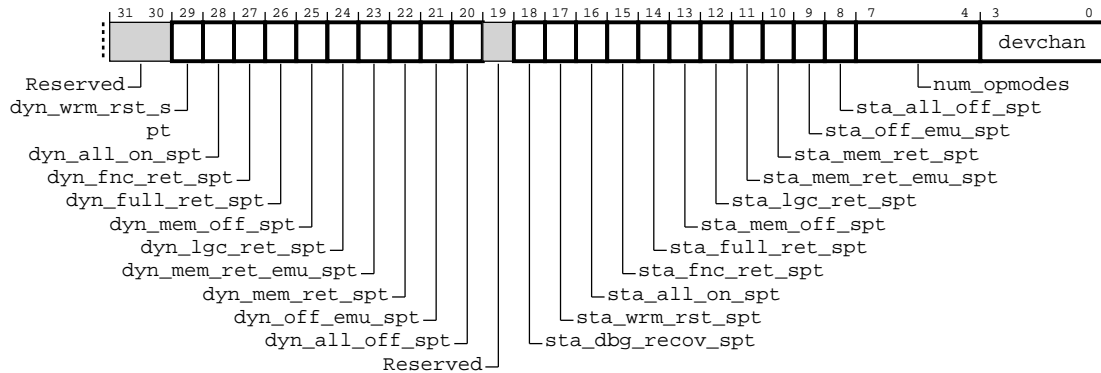


Table 5-166: cmn_hns_ppu_idr0 attributes

Bits	Name	Description	Type	Reset
[31:30]	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

Bits	Name	Description	Type	Reset
[8]	sta_all_off_spt	Static off support	RO	1'b1
[7:4]	num_opmodes	Number of operational modes	RO	4'b0100
[3:0]	devchan	Number of device interface channels	RO	1'b0

5.2.4.19 cmn_hns_ppu_idr1

Provides identification information for the HN-F PPU.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h28B4

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-163: cmn_hns_ppu_idr1

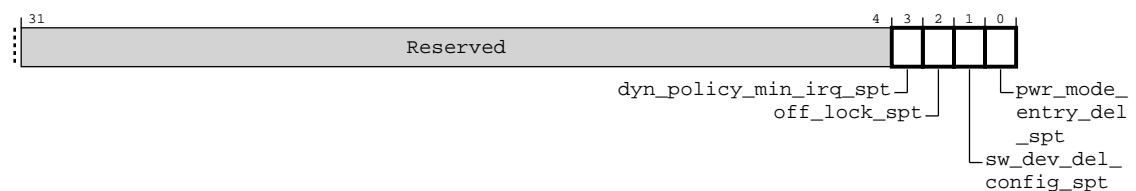


Table 5-167: cmn_hns_ppu_idr1 attributes

Bits	Name	Description	Type	Reset
[31:4]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[3]	dyn_policy_min_irq_spt	Dynamic minimum policy interrupt support	RO	1'b0
[2]	off_lock_spt	Off and mem_ret lock support	RO	1'b0
[1]	sw_dev_del_config_spt	Software device delay control configuration support	RO	1'b0
[0]	pwr_mode_entry_del_spt	Power mode entry delay support	RO	1'b0

5.2.4.20 cmn_hns_ppu_iidr

Functions as the power implementation identification register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h28C8

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-164: cmn_hns_ppu_iidr

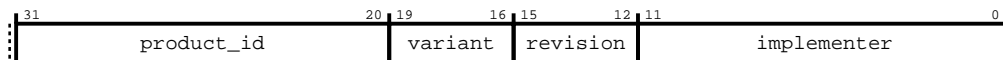


Table 5-168: cmn_hns_ppu_iidr attributes

Bits	Name	Description	Type	Reset
[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

Bits	Name	Description	Type	Reset
[11:0]	implementer	Arm implementation	RO	12'h43B

5.2.4.21 cmn_hns_ppu_aidr

Functions as the power architecture identification register for HN-F.

Configurations

This register is available in all configurations.

Attributes

Width

32

Address offset

16'h28CC

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-165: cmn_hns_ppu_aidr

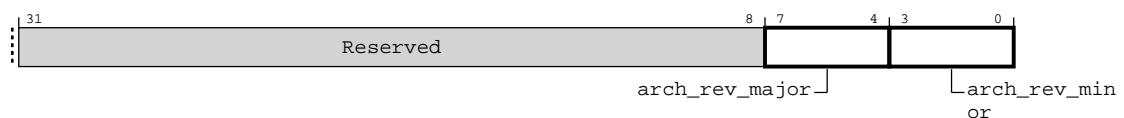


Table 5-169: cmn_hns_ppu_aidr attributes

Bits	Name	Description	Type	Reset
[31:8]	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

5.2.4.22 cmn_hns_ppu_dyn_ret_threshold

Configures the dynamic retention threshold for SLC and SF RAM.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1A00

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ppu

Secure group override

cmn_hns_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ppu bit and cmn_hns_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-166: cmn_hns_ppu_dyn_ret_threshold

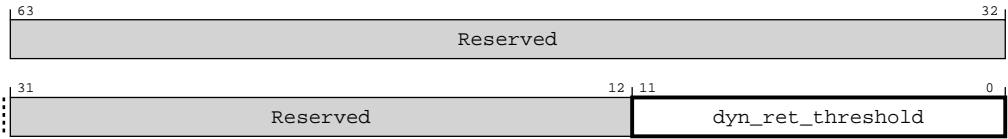


Table 5-170: cmn_hns_ppu_dyn_ret_threshold attributes

Bits	Name	Description	Type	Reset
[63:12]	Reserved	Reserved	RO	-
[11:0]	dyn_ret_threshold	HN-F RAM idle cycle count threshold	RW	32'b0

5.2.4.23 cmn_hns_qos_band

Provides QoS classifications based on the QoS value ranges.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hA80

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-167: cmn_hns_qos_band

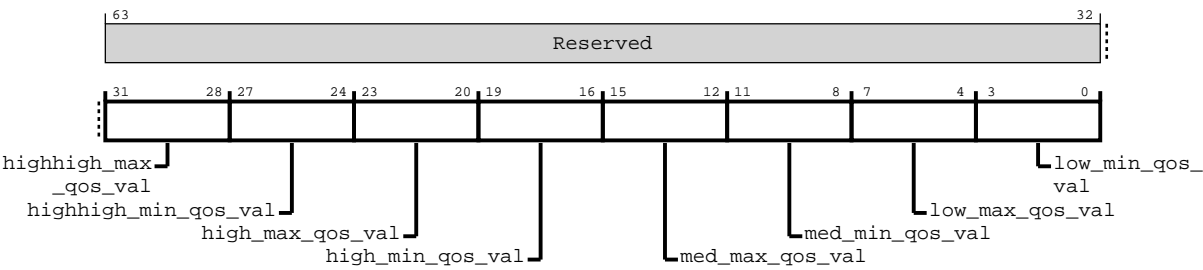


Table 5-171: cmn_hns_qos_band attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	highhigh_max_qos_val	Maximum value for HighHigh QoS class	RO	4'hF

Bits	Name	Description	Type	Reset
[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

5.2.4.24 cmn_hns_errfr

Functions as the error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE000

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-168: cmn_hns_errfr

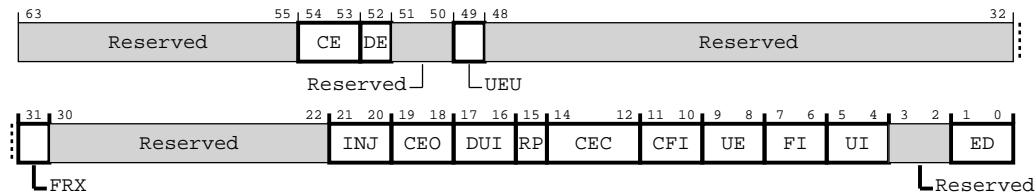


Table 5-172: cmn_hns_errfr attributes

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-
[54:53]	CE	Corrected Error recording 2'b00 Corrected Error not supported 2'b10 Non-specific Corrected Error supported	RO	2'b10
[52]	DE	Deferred Error recording 1'b0 Deferred Error not supported 1'b1 Deferred Error supported	RO	1'b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording 1'b0: Unrecoverable Error not supported 1'b1: Unrecoverable Error supported	RO	1'b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension. 1'b1: cmn_hns_errfr[63:48] is architecturally defined	RO	1'b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension. 2'b01: Support Common Fault Injection Model Extension	RO	2'b01
[19:18]	CEO	Corrected Error overwrite. 2'b00: Keep the first Corrected Error syndrome	RO	2'b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control 2'b00: Does not support Error Recovery Interrupt from Deferred errors 2'b10: Support Error Recovery Interrupt from Deferred errors and controllable using cmn_hns_errctlr.DUI.	RO	2'b10
[15]	RP	Repeat counter (valid only when cmn_hns_errfr.CEC != 3'b000.) 1'b0: Invalid; 1'b1: Implements a first (repeat) counter and a second (other) counter in cmn_hns_errmisc0	RO	1'b1
[14:12]	CEC	Standard corrected error counter 3'b000: Does not implement standard error counter model 3'b100: Implements a 16-bit Corrected error counter in cmn_hns_errmisc0	RO	3'b100
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control 2'b00: Does not support Fault Handling Interrupt from Corrected errors 2'b10: Support Fault Handling Interrupt on corrected errors and controllable using cmn_hns_errctlr.CFI.	RO	2'b10
[9:8]	UE	In-band error response is always on	RO	2'b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control 2'b00: Does not support Fault Handling Interrupt from Deferred and Uncorrected errors 2'b10: Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using cmn_hns_errctlr.FI.	RO	2'b10

Bits	Name	Description	Type	Reset
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control 2'b00: Does not support Error Recovery Interrupt from Uncorrected errors 2'b10: Support Error Recovery Interrupt on Uncorrected errors and controllable using cmn_hns_errctlr.UI.	RO	2'b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control 2'b10: Error reporting and logging is controllable using cmn_hns_errctlr.ED	RO	2'b10

5.2.4.25 cmn_hns_errctlr

Functions as the error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE008

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ras bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ras bit and cmn_hns_scr.ras bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-169: cmn_hns_errctlr

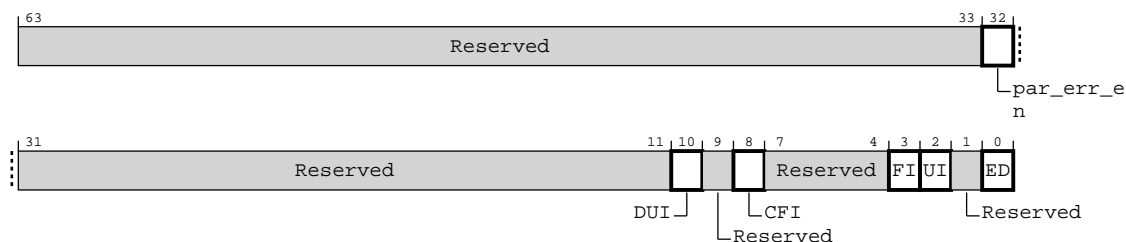


Table 5-173: cmn_hns_errctlr attributes

Bits	Name	Description	Type	Reset
[63:33]	Reserved	Reserved	RO	-
[32]	par_err_en	Enables external logging parity errors when set to 1'b1	RW	1'b0
[31:11]	Reserved	Reserved	RO	-
[10]	DUI	Enables error recovery interrupt for deferred error as specified in cmn_hns_errfr.DUI	RW	1'b0
[9]	Reserved	Reserved	RO	-
[8]	CFI	Enables fault handling interrupt for corrected error as specified in cmn_hns_errfr.CFI	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in cmn_hns_errfr.FI	RW	1'b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in cmn_hns_errfr.UI	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	ED	Enables error detection as specified in cmn_hns_errfr.ED	RW	1'b0

5.2.4.26 cmn_hns_errstatus

Functions as the error status register. When V is set, only write exact same value as in the register can clear it.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE010

Type

W1C

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ras bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ras bit and cmn_hns_scr.ras bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-170: cmn_hns_errstatus

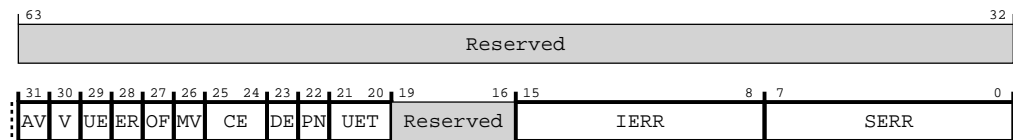


Table 5-174: cmn_hns_errstatus attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid 1'b1: Address is valid; cmn_hns_erraddr contains a physical address for that recorded error 1'b0: Address is not valid	W1C	1'b0
[30]	V	Status register valid 1'b1: At least one error recorded; register is valid 1'b0: No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors 1'b1: At least one error detected that is not corrected and is not deferred to a subordinate 1'b0: No uncorrected errors detected	W1C	1'b0
[28]	ER	Error Reported 1'b1: In-band error response signaled to the Requester 1'b0: No in-band error response signaled	W1C	1'b0
[27]	OF	Overflow; asserted when multiple errors are detected 1'b1: More than one error detected 1'b0: None or only one error detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	cmn_hns_errmisc<01> valid 1'b1: Miscellaneous registers are valid 1'b0: Miscellaneous registers are not valid	W1C	1'b0
[25:24]	CE	Corrected errors 2'b10: At least one corrected error recorded 2'b00: No corrected errors recorded	W1C	2'b00
[23]	DE	Deferred errors 1'b1: At least one error is not corrected and is deferred 1'b0: No errors deferred	W1C	1'b0
[22]	PN	Poison 1'b1: Uncorrected error recorded because a poison value was consumed 1'b0: Other cases	W1C	1'b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0 2'b01: Uncorrected error, Unrecoverable error (UEU). 2'b00: Invalid	W1C	2'b00
[19:16]	Reserved	Reserved	RO	-
[15:8]	IERR	Implementation-defined primary error code. 8'h00: No error 8'h01: Partner implementation defined error	W1C	8'b0

Bits	Name	Description	Type	Reset
[7:0]	SERR	Architecturally-defined primary error code. 8'h00: No error 8'h01: CMN implementation defined error. Refer to cmn_hns_errmisc1.ERRSRC for error type details. 8'h06: ECC error on L3 data 8'h07: ECC error on L3/SF Tag 8'h0A: Producer write data was poisoned but ACE-Lite does not support poisoned data 8'h0D: Illegal address 8'h1A: Parity error	W1C	8'b0

5.2.4.27 cmn_hns_erraddr

Contains the error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE018

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ras bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ras bit and cmn_hns_scr.ras bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-171: cmn_hns_erraddr

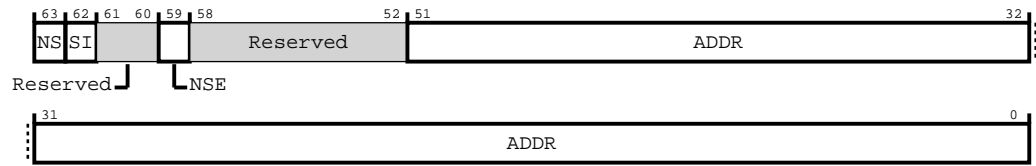


Table 5-175: cmn_hns_erraddr attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	1'b0
[62]	SI	{NSE,NS} valid 1'b0: PAS field is valid 1'b1: PAS field is invalid	RW	1'b0
[61:60]	Reserved	Reserved	RO	-
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	1'b0
[58:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

5.2.4.28 cmn_hns_errmisc0

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'hE020

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras_secure_access_override

Secure group override

cmn_hns_scr.ras_secure_access_override

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.ras_secure_access_override` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.ras_secure_access_override` bit and `cmn_hns_scr.ras_secure_access_override` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-172: `cmn_hns_errmisc0`

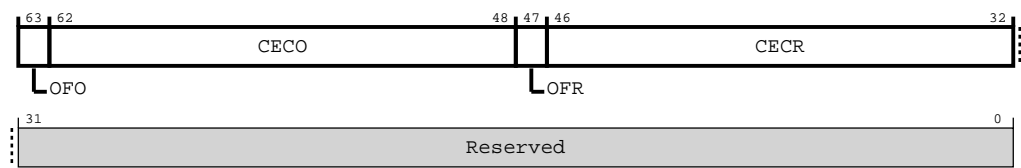


Table 5-176: `cmn_hns_errmisc0` attributes

Bits	Name	Description	Type	Reset
[63]	OFO	Corrected error counter overflow	RW	1'b0
[62:48]	CECO	Corrected ECC error count	RW	15'b0
[47]	OFR	Corrected error counter overflow	RW	1'b0
[46:32]	CECR	Corrected ECC error count	RW	15'b0
[31:0]	Reserved	Reserved	RO	-

5.2.4.29 `cmn_hns_errmisc1`

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'hE028

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras_secure_access_override

Secure group override

cmn_hns_scr.ras_secure_access_override

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ras_secure_access_override bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ras_secure_access_override bit and cmn_hns_scr.ras_secure_access_override bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-173: cmn_hns_errmisc1

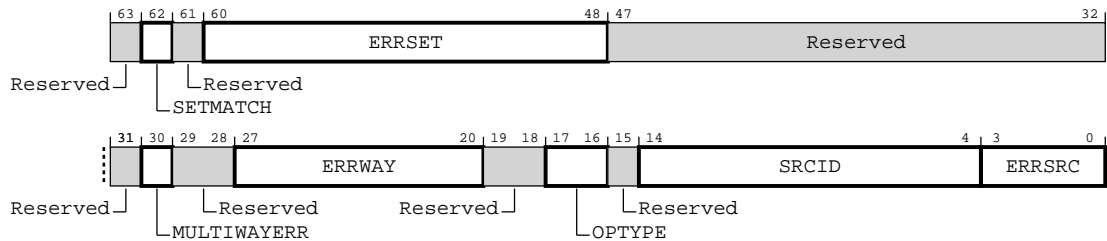


Table 5-177: cmn_hns_errmisc1 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	SETMATCH	Set address match	RW	1'b0
[61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	SLC/SF set address for ECC error	RW	13'b0
[47:31]	Reserved	Reserved	RO	-
[30]	MULTIWAYERR	Indicate multiple ways have ECC error	RW	1'b0
[29:28]	Reserved	Reserved	RO	-
[27:20]	ERRWAY	SLC/SF way ID for ECC error	RW	8'b0
[19:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error op type 2'b00: Writes, CleanShared, Atomics and stash requests with invalid targets 2'b01: WriteBack, Evict, and Stash requests with valid target 2'b10: CMO 2'b11: Other op types	RW	2'b00
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0

Bits	Name	Description	Type	Reset
[3:0]	ERRSRC	Error source 4'b0001: Data single-bit ECC 4'b0010: Data double-bit ECC 4'b0011: Single-bit ECC overflow 4'b0100: Tag single-bit ECC 4'b0101: Tag double-bit ECC 4'b0111: SF tag single-bit ECC 4'b1000: SF tag double-bit ECC 4'b1010: Data parity error 4'b1011: Data parity and poison 4'b1100: NDE	RW	4'b0000

5.2.4.30 cmn_hns_errpfgf

Functions as the Pseudo-fault Generation Feature Register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE800

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-174: cmn_hns_errpfgf

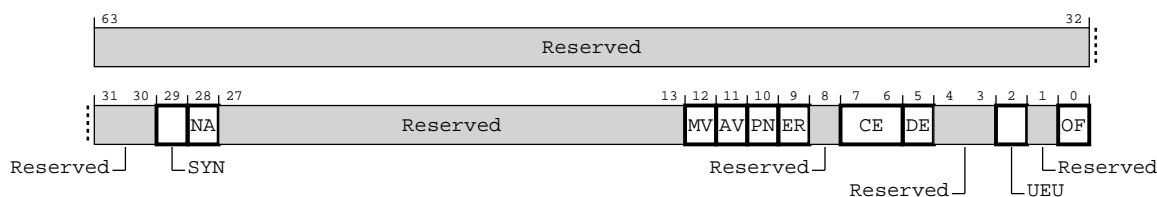


Table 5-178: cmn_hns_errpfgf attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29]	SYN	Fault syndrome injection. 1'b1: Fault injection does not update ERRSTATUS.SERR	RO	1'b1
[28]	NA	No access required. 1'b1: Fault injection does not require cfg access	RO	1'b1
[27:13]	Reserved	Reserved	RO	-
[12]	MV	Miscellaneous syndrome. 1'b1: Fault injection update ERRSTATUS.MV to ERRPFGCTL.MV	RO	1'b1
[11]	AV	Address syndrome. 1'b1: Fault injection update ERRSTATUS.AV to ERRPFGCTL.AV	RO	1'b1
[10]	PN	Poison flag 1'b1: Fault injection update ERRSTATUS.PN to ERRPFGCTL.PN	RO	1'b1
[9]	ER	Error reported flag 1'b1: Fault injection update ERRSTATUS.ER to ERRPFGCTL.ER	RO	1'b1
[8]	Reserved	Reserved	RO	-
[7:6]	CE	Corrected Error generation. 2'b00: No Corrected error generation. 2'b01: Non specific Corrected error injection. If ERRPFGCTL.CE == 1, update ERRSTATUS.CE to 2'b10; else, update ERRSTATUS.CE to 2'b00	RO	2'b01
[5]	DE	Deferred error generation. 1'b0: No Deferred error generation. 1'b1: Fault injection update ERRSTATUS.DE to ERRPFGCTL.DE	RO	1'b1
[4:3]	Reserved	Reserved	RO	-
[2]	UEU	Uncorrected error generation. 1'b1: If ERRPFGCTL.UEU == 1, update ERRSTATUS.UE to 1'b1 and ERRSTATUS.UET = 2'b01; else, update ERRSTATUS.UE to 1'b0 and ERRSTATUS.UET = 2'b00	RO	1'b1
[1]	Reserved	Reserved	RO	-
[0]	OF	Overflow flag. 1'b1: Fault injection update ERRSTATUS.OF to ERRPFGCTL.OF	RO	1'b1

5.2.4.31 cmn_hns_errpfgctl

Functions as the Pseudo-fault Generation Control Register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE808

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ras bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ras bit and cmn_hns_scr.ras bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-175: cmn_hns_errpfgctl

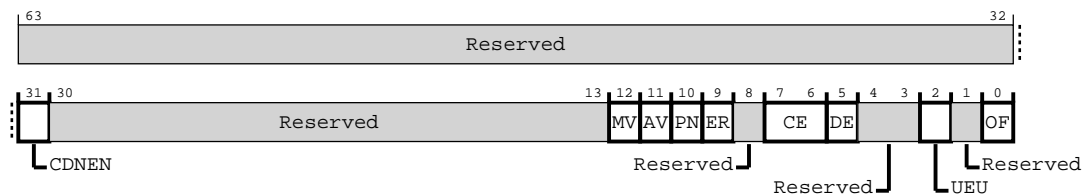


Table 5-179: cmn_hns_errpfgctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	CDNEN	Countdown Enable. 1'b0: Countdown disabled. 1'b1: Error generation counter is set to ERRPFGCDN.CDN, and countdown enabled.	RW	1'b0
[30:13]	Reserved	Reserved	RO	-
[12]	MV	Miscellaneous syndrome. 1'b0: Fault injection update ERRSTATUS.MV to 1'b0 1'b1: Fault injection update ERRSTATUS.MV to 1'b1	RW	1'b0
[11]	AV	Address syndrome. 1'b0: Fault injection update ERRSTATUS.AV to 1'b0 1'b1: Fault injection update ERRSTATUS.AV to 1'b1	RW	1'b0
[10]	PN	Poison flag 1'b0: Fault injection update ERRSTATUS.PN to 1'b0 1'b1: Fault injection update ERRSTATUS.PN to 1'b1	RW	1'b0
[9]	ER	Error reported flag 1'b0: Fault injection update ERRSTATUS.ER to 1'b0 1'b1: Fault injection update ERRSTATUS.ER to 1'b1	RW	1'b0
[8]	Reserved	Reserved	RO	-
[7:6]	CE	Corrected Error generation. 2'b00: Non Corrected error is injected. Fault injection update ERRSTATUS.CE to 2'b00 2'b01: Non specific Corrected error injection. Fault injection update ERRSTATUS.CE to 2'b10	RW	2'b00
[5]	DE	Deferred error generation. 1'b0: Fault injection update ERRSTATUS.DE to 1'b0 1'b1: Fault injection update ERRSTATUS.DE to 1'b1	RW	1'b0
[4:3]	Reserved	Reserved	RO	-
[2]	UEU	Uncorrected error generation. 1'b0: Fault injection update ERRSTATUS.UE to 1'b0 and ERRSTATUS.UET = 2'b00 1'b1: Fault injection update ERRSTATUS.UE to 1'b1 and ERRSTATUS.UET = 2'b01	RW	1'b0
[1]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[0]	OF	Overflow flag. 1'b0: Fault injection update ERRSTATUS.OF to 1'b0 1'b1: Fault injection update ERRSTATUS.OF to 1'b1	RW	1'b0

5.2.4.32 cmn_hns_errpfgcdn

Functions as the Pseudo-fault Generation Countdown Register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE810

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ras bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ras bit and cmn_hns_scr.ras bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-176: cmn_hns_errpfgcdn

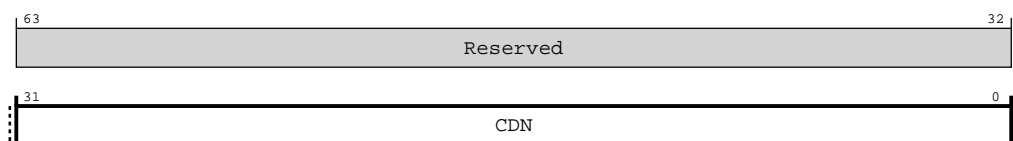


Table 5-180: cmn_hns_errpfgcdn attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	CDN	Countdown value	RW	32'b0

5.2.4.33 cmn_hns_errfr_NS

Functions as the non-secure error feature register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE040

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-177: cmn_hns_errfr_NS

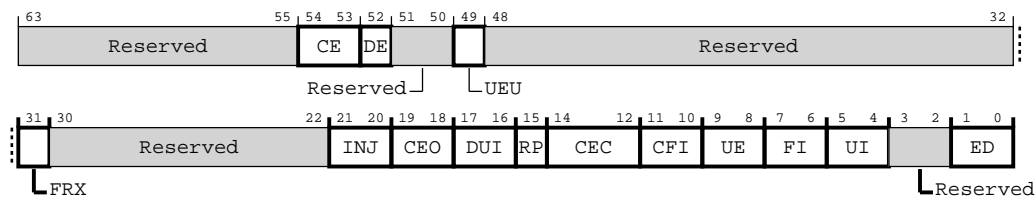


Table 5-181: cmn_hns_errfr_NS attributes

Bits	Name	Description	Type	Reset
[63:55]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[54:53]	CE	Corrected Error recording 2'b00 Corrected Error not supported 2'b10 Non-specific Corrected Error supported	RO	2'b10
[52]	DE	Deferred Error recording 1'b0 Deferred Error not supported 1'b1 Deferred Error supported	RO	1'b1
[51:50]	Reserved	Reserved	RO	-
[49]	UEU	Unrecoverable Error recording 1'b0: Unrecoverable Error not supported 1'b1: Unrecoverable Error supported	RO	1'b1
[48:32]	Reserved	Reserved	RO	-
[31]	FRX	Feature Register extension. 1'b1: cmn_hns_errfr_NS[63:48] is architecturally defined	RO	1'b1
[30:22]	Reserved	Reserved	RO	-
[21:20]	INJ	Fault Injection Extension. 2'b01: Support Common Fault Injection Model Extension	RO	2'b01
[19:18]	CEO	Corrected Error overwrite. 2'b00: Keep the first Corrected Error syndrome	RO	2'b00
[17:16]	DUI	Error Recovery Interrupt from Deferred errors control 2'b00: Does not support Error Recovery Interrupt from Deferred errors 2'b10: Support Error Recovery Interrupt from Deferred errors and controllable using cmn_hns_errctlr.DUI.	RO	2'b10
[15]	RP	Repeat counter (valid only when cmn_hns_errfr_NS.CEC != 3'b000.) 1'b1: Implements a first (repeat) counter and a second (other) counter in cmn_hns_errmisc0	RO	1'b1
[14:12]	CEC	Standard corrected error counter 3'b000: Does not implement standard error counter model 3'b100: Implements a 16-bit Corrected error counter in cmn_hns_errmisc0	RO	3'b100
[11:10]	CFI	Fault Handling Interrupt from Corrected errors control 2'b00: Does not support Fault Handling Interrupt from Corrected errors 2'b10: Support Fault Handling Interrupt on corrected errors and controllable using cmn_hns_errctlr.CFI.	RO	2'b10
[9:8]	UE	In-band error response is always on	RO	2'b01
[7:6]	FI	Fault Handling Interrupt from Deferred and Uncorrected errors control 2'b00: Does not support Fault Handling Interrupt from Deferred and Uncorrected errors 2'b10: Support Fault Handling Interrupt on Deferred and Uncorrected errors and controllable using cmn_hns_errctlr.FI.	RO	2'b10
[5:4]	UI	Error Recovery Interrupt from Uncorrected errors control 2'b00: Does not support Error Recovery Interrupt from Uncorrected errors 2'b10: Support Error Recovery Interrupt on Uncorrected errors and controllable using cmn_hns_errctlr.UI.	RO	2'b10
[3:2]	Reserved	Reserved	RO	-
[1:0]	ED	Error reporting and logging control 2'b10: Error reporting and logging is controllable using cmn_hns_errctlr.ED	RO	2'b10

5.2.4.34 cmn_hns_errctlr_NS

Functions as the non-secure error control register. Controls whether specific error-handling interrupts and error detection/deferment are enabled.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE048

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-178: cmn_hns_errctlr_NS

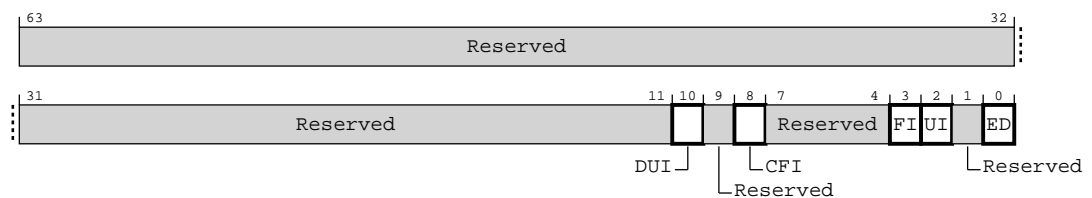


Table 5-182: cmn_hns_errctlr_NS attributes

Bits	Name	Description	Type	Reset
[63:11]	Reserved	Reserved	RO	-
[10]	DUI	Enables error recovery interrupt for deferred error as specified in <code>cmn_hns_errfr_NS.DUI</code>	RW	1'b0
[9]	Reserved	Reserved	RO	-
[8]	CFI	Enables fault handling interrupt for corrected error as specified in <code>cmn_hns_errfr_NS.CFI</code>	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	FI	Enables fault handling interrupt for uncorrected and deferred errors as specified in <code>cmn_hns_errfr_NS.FI</code>	RW	1'b0
[2]	UI	Enables error recovery interrupt for uncorrected error as specified in <code>cmn_hns_errfr_NS.UI</code>	RW	1'b0
[1]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[0]	ED	Enables error detection as specified in cmn_hns_errfr_NS.ED	RW	1'b0

5.2.4.35 cmn_hns_errstatus_NS

Functions as the non-secure error status register. When V is set, only write exact same value as in the register can clear it.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE050

Type

W1C

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-179: cmn_hns_errstatus_NS

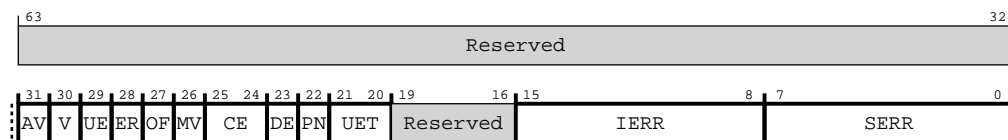


Table 5-183: cmn_hns_errstatus_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	AV	Address register valid 1'b1: Address is valid; cmn_hns_erraddr contains a physical address for that recorded error 1'b0: Address is not valid	W1C	1'b0
[30]	V	Status register valid 1'b1: At least one error recorded; register is valid 1'b0: No errors recorded	W1C	1'b0
[29]	UE	Uncorrected errors 1'b1: At least one error detected that is not corrected and is not deferred to a subordinate 1'b0: No uncorrected errors detected	W1C	1'b0

Bits	Name	Description	Type	Reset
[28]	ER	Error Reported 1'b1: In-band error response signaled to the Requester 1'b0: No in-band error response signaled	W1C	1'b0
[27]	OF	Overflow; asserted when multiple errors are detected 1'b1: More than one error detected 1'b0: None or only one error detected as described by UE/DE/CE fields	W1C	1'b0
[26]	MV	cmn_hns_errmisc<01> valid 1'b1: Miscellaneous registers are valid 1'b0: Miscellaneous registers are not valid	W1C	1'b0
[25:24]	CE	Corrected errors 2'b10: At least one corrected error recorded 2'b00: No corrected errors recorded	W1C	2'b00
[23]	DE	Deferred errors 1'b1: At least one error is not corrected and is deferred 1'b0: No errors deferred	W1C	1'b0
[22]	PN	Poison 1'b1: Uncorrected error recorded because a poison value was consumed 1'b0: Other cases	W1C	1'b0
[21:20]	UET	Uncorrected Error Type, valid only when UE != 0 2'b01: Uncorrected error, Unrecoverable error (UEU). 2'b00: Invalid	W1C	2'b00
[19:16]	Reserved	Reserved	RO	-
[15:8]	IERR	Implementation-defined primary error code. 8'h00: No error 8'h01: Partner implementation defined error	W1C	8'b0
[7:0]	SERR	Architecturally-defined primary error code. 8'h00: No error 8'h01: CMN implementation defined error. Refer to cmn_hns_errmisc1_NS.ERRSRC for error type details. 8'h06: ECC error on cache data 8'h07: ECC error on L3/SF Tag 8'h0A: Producer write data was poisoned but ACE-Lite does not support poisoned data 8'h0D: Illegal address 8'h1A: Parity error	W1C	8'b0

5.2.4.36 cmn_hns_erraddr_NS

Contains the non-secure error record address.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE058

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-180: cmn_hns_erraddr_NS

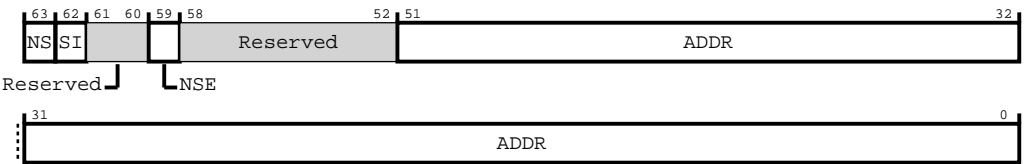


Table 5-184: cmn_hns_erraddr_NS attributes

Bits	Name	Description	Type	Reset
[63]	NS	Security status of transaction. PAS[0] of the transaction.	RW	1'b0
[62]	SI	{NSE,NS} valid 1'b0: PAS field is valid 1'b1: PAS field is invalid	RW	1'b0
[61:60]	Reserved	Reserved	RO	-
[59]	NSE	Root status of transaction. PAS[1] of the transaction.	RW	1'b0
[58:52]	Reserved	Reserved	RO	-
[51:0]	ADDR	Transaction address	RW	52'b0

5.2.4.37 cmn_hns_errmisc0_NS

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'hE060

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-181: cmn_hns_errmisc0_NS

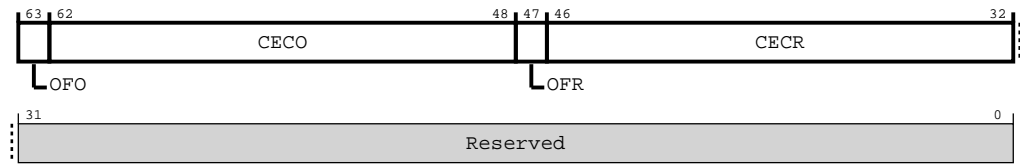


Table 5-185: cmn_hns_errmisc0_NS attributes

Bits	Name	Description	Type	Reset
[63]	OFO	Corrected error counter overflow	RW	1'b0
[62:48]	CECO	Corrected ECC error count	RW	15'b0
[47]	OFR	Corrected error counter overflow	RW	1'b0
[46:32]	CECR	Corrected ECC error count	RW	15'b0
[31:0]	Reserved	Reserved	RO	-

5.2.4.38 cmn_hns_errmisc1_NS

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'hE068

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-182: cmn_hns_errmisc1_NS

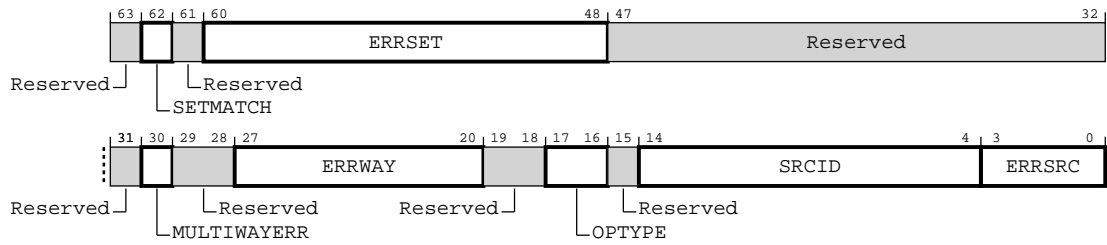


Table 5-186: cmn_hns_errmisc1_NS attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62]	SETMATCH	Set address match	RW	1'b0
[61]	Reserved	Reserved	RO	-
[60:48]	ERRSET	SLC/SF set address for ECC error	RW	13'b0
[47:31]	Reserved	Reserved	RO	-
[30]	MULTIWAYERR	Indicate multiple ways have ECC error	RW	1'b0
[29:28]	Reserved	Reserved	RO	-
[27:20]	ERRWAY	SLC/SF way ID for ECC error	RW	8'b0
[19:18]	Reserved	Reserved	RO	-
[17:16]	OPTYPE	Error op type 2'b00: Writes, CleanShared, Atomics and stash requests with invalid targets 2'b01: WriteBack, Evict, and Stash requests with valid target 2'b10: CMO 2'b11: Other op types	RW	2'b00
[15]	Reserved	Reserved	RO	-
[14:4]	SRCID	Error source ID	RW	11'b0
[3:0]	ERRSRC	Error source 4'b0001: Data single-bit ECC 4'b0010: Data double-bit ECC 4'b0011: Single-bit ECC overflow 4'b0100: Tag single-bit ECC 4'b0101: Tag double-bit ECC 4'b0111: SF tag single-bit ECC 4'b1000: SF tag double-bit ECC 4'b1010: Data parity error 4'b1011: Data parity and poison 4'b1100: NDE	RW	4'b0000

5.2.4.39 cmn_hns_errpfgf_NS

Functions as the non-secure Pseudo-fault Generation Feature Register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE840

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-183: cmn_hns_errpfgf_NS

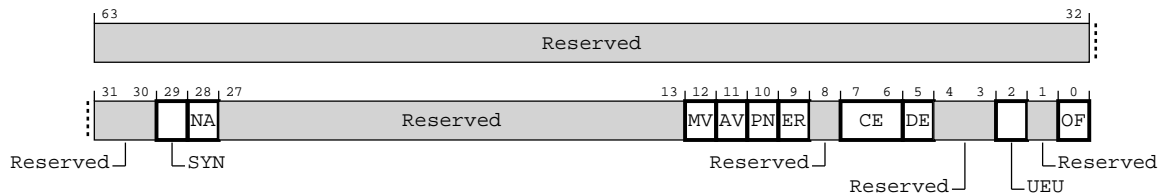


Table 5-187: cmn_hns_errpfgf_NS attributes

Bits	Name	Description	Type	Reset
[63:30]	Reserved	Reserved	RO	-
[29]	SYN	Fault syndrome injection. 1'b1: Fault injection does not update ERRSTATUS_NS.SERR	RO	1'b1
[28]	NA	No access required. 1'b1: Fault injection does not require cfg access	RO	1'b1
[27:13]	Reserved	Reserved	RO	-
[12]	MV	Miscellaneous syndrome. 1'b1: Fault injection update ERRSTATUS_NS.MV to ERRPFGCTL_NS.MV	RO	1'b1
[11]	AV	Address syndrome. 1'b1: Fault injection update ERRSTATUS_NS.AV to ERRPFGCTL_NS.AV	RO	1'b1
[10]	PN	Poison flag 1'b1: Fault injection update ERRSTATUS_NS.PN to ERRPFGCTL_NS.PN	RO	1'b1
[9]	ER	Error reported flag 1'b1: Fault injection update ERRSTATUS_NS.ER to ERRPFGCTL_NS.ER	RO	1'b1
[8]	Reserved	Reserved	RO	-
[7:6]	CE	Corrected Error generation. 2'b00: No Corrected error generation. 2'b01: Non specific Corrected error injection. If ERRPFGCTL_NS.CE == 1, update ERRSTATUS_NS.CE to 2'b10; else, update ERRSTATUS_NS.CE to 2'b00	RO	2'b01
[5]	DE	Deferred error generation. 1'b0: No Deferred error generation. 1'b1: Fault injection update ERRSTATUS_NS.DE to ERRPFGCTL_NS.DE	RO	1'b1
[4:3]	Reserved	Reserved	RO	-
[2]	UEU	Uncorrected error generation. 1'b1: If ERRPFGCTL_NS.UEU == 1, update ERRSTATUS_NS.UE to 1'b1 and ERRSTATUS_NS.UET = 2'b01; else, update ERRSTATUS_NS.UE to 1'b0 and ERRSTATUS_NS.UET = 2'b00	RO	1'b1
[1]	Reserved	Reserved	RO	-
[0]	OF	Overflow flag. 1'b1: Fault injection update ERRSTATUS_NS.OF to ERRPFGCTL_NS.OF	RO	1'b1

5.2.4.40 cmn_hns_errpfgctl_NS

Functions as the non-secure Pseudo-fault Generation Control Register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE848

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-184: cmn_hns_errpfgctl_NS

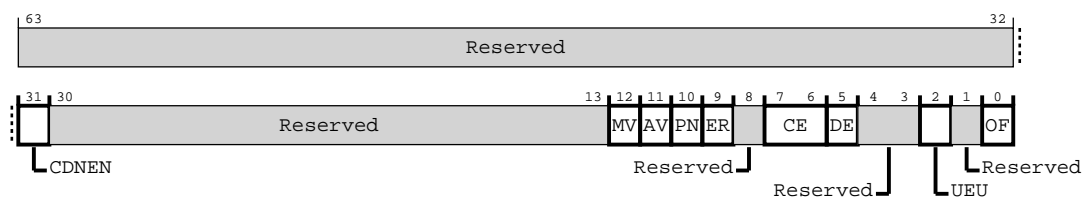


Table 5-188: cmn_hns_errpfgctl_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31]	CDNEN	Countdown Enable. 1'b0: Countdown disabled. 1'b1: Error generation counter is set to ERRPFGCDN_NS.CDN, and countdown enabled.	RW	1'b0
[30:13]	Reserved	Reserved	RO	-
[12]	MV	Miscellaneous syndrome. 1'b0: Fault injection update ERRSTATUS_NS.MV to 1'b0 1'b1: Fault injection update ERRSTATUS_NS.MV to 1'b1	RW	1'b0
[11]	AV	Address syndrome. 1'b0: Fault injection update ERRSTATUS_NS.AV to 1'b0 1'b1: Fault injection update ERRSTATUS_NS.AV to 1'b1	RW	1'b0
[10]	PN	Poison flag 1'b0: Fault injection update ERRSTATUS_NS.PN to 1'b0 1'b1: Fault injection update ERRSTATUS_NS.PN to 1'b1	RW	1'b0

Bits	Name	Description	Type	Reset
[9]	ER	Error reported flag 1'b0: Fault injection update ERRSTATUS_NS.ER to 1'b0 1'b1: Fault injection update ERRSTATUS_NS.ER to 1'b1	RW	1'b0
[8]	Reserved	Reserved	RO	-
[7:6]	CE	Corrected Error generation. 2'b00: Non Corrected error is injected. Fault injection update ERRSTATUS_NS.CE to 2'b00 2'b01: Non specific Corrected error injection. Fault injection update ERRSTATUS_NS.CE to 2'b10	RW	2'b00
[5]	DE	Deferred error generation. 1'b0: Fault injection update ERRSTATUS_NS.DE to 1'b0 1'b1: Fault injection update ERRSTATUS_NS.DE to 1'b1	RW	1'b0
[4:3]	Reserved	Reserved	RO	-
[2]	UEU	Uncorrected error generation. 1'b0: Fault injection update ERRSTATUS_NS.UE to 1'b0 and ERRSTATUS_NS.UET = 2'b00 1'b1: Fault injection update ERRSTATUS_NS.UE to 1'b1 and ERRSTATUS_NS.UET = 2'b01	RW	1'b0
[1]	Reserved	Reserved	RO	-
[0]	OF	Overflow flag. 1'b0: Fault injection update ERRSTATUS_NS.OF to 1'b0 1'b1: Fault injection update ERRSTATUS_NS.OF to 1'b1	RW	1'b0

5.2.4.41 cmn_hns_errpfgcdn_NS

Functions as the non-secure Pseudo-fault Generation Countdown Register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE850

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-185: cmn_hns_errpfgcdn_NS

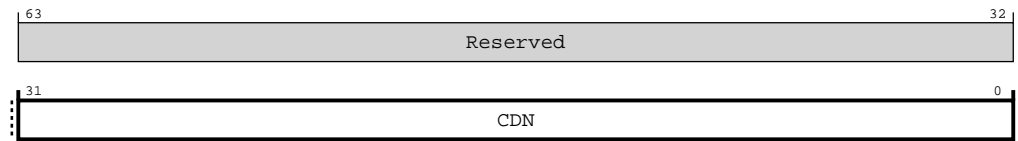


Table 5-189: cmn_hns_errpfgcdn_NS attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	CDN	Countdown value	RW	32'b0

5.2.4.42 cmn_hns_errcapctl

Functions as the error Capture Control Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hED00

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ras

Secure group override

cmn_hns_scr.ras

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.ras` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.ras` bit and `cmn_hns_scr.ras` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-186: cmn_hns_errcapctl

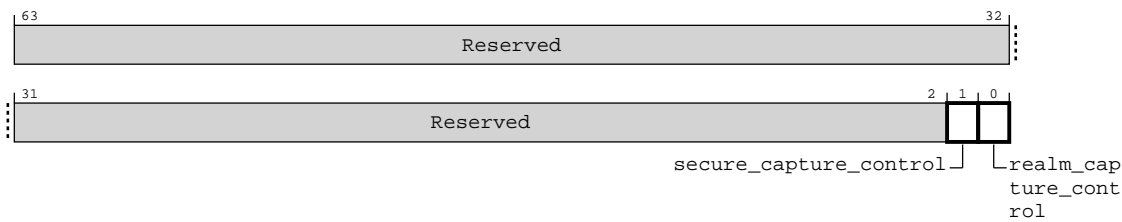


Table 5-190: cmn_hns_errcapctl attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	secure_capture_control	Secure Capture Control 1'b0: Transaction with secure PAS captured in root error record 1'b1: Transaction with secure PAS captured in non-secure error record	RW	1'b0
[0]	realm_capture_control	Realm Capture Control 1'b0: Transaction with realm PAS captured in root error record 1'b1: Transaction with realm PAS captured in non-secure error record	RW	1'b0

5.2.4.43 cmn_hns_errgsr

Functions as Error Group Status Register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEE00

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-187: cmn_hns_errgsr

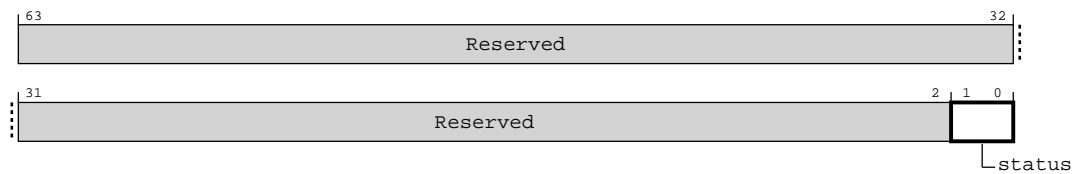


Table 5-191: cmn_hns_errgsr attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	status	Read-only copy of {ERR<n>STATUS_NS.V, ERR<n>STATUS.V}	RO	2'b0

5.2.4.44 cmn_hns_erridr

Functions as the implementation identification register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEE10

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-188: cmn_hns_erridr

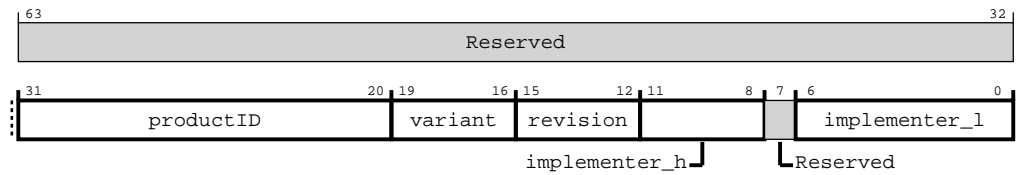


Table 5-192: cmn_hns_erridr attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:20]	productID	Product Part number	RO	12'h0
[19:16]	variant	Component major revision	RO	4'h0
[15:12]	revision	Component minor revision	RO	4'h0
[11:8]	implementer_h	Implementer[10:7]	RO	4'h4
[7]	Reserved	Reserved	RO	-
[6:0]	implementer_l	Implementer[6:0]	RO	7'h3B

5.2.4.45 cmn_hns_errdevaff

Functions as the device affinity register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEFA8

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-189: cmn_hns_errdevaff

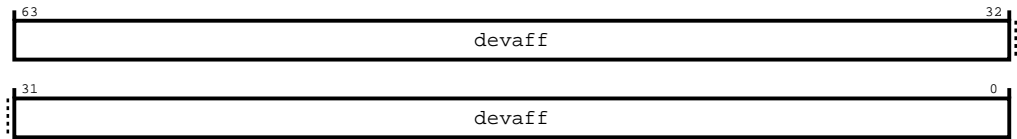


Table 5-193: cmn_hns_errdevaff attributes

Bits	Name	Description	Type	Reset
[63:0]	devaff	Device affinity register	RO	64'b0

5.2.4.46 cmn_hns_errdevarch

Functions as the device architecture register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEFB8

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-190: cmn_hns_errdevarch

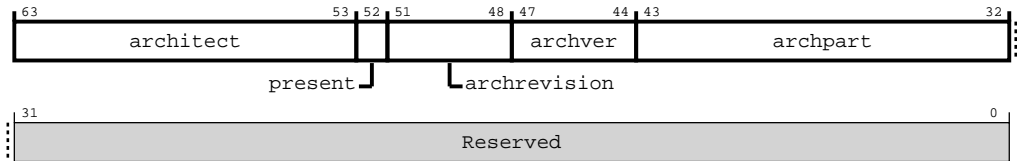


Table 5-194: cmn_hns_errdevarch attributes

Bits	Name	Description	Type	Reset
[63:53]	architect	Architect	RO	11'h23B
[52]	present	Present	RO	1'b1
[51:48]	archrevision	Architecture revision	RO	4'b1
[47:44]	archver	Architecture Version	RO	4'h0
[43:32]	archpart	Architecture Part	RO	12'hA00
[31:0]	Reserved	Reserved	RO	-

5.2.4.47 cmn_hns_errdev

Functions as the device configuration register

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hEFC8

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-191: cmn_hns_errdev

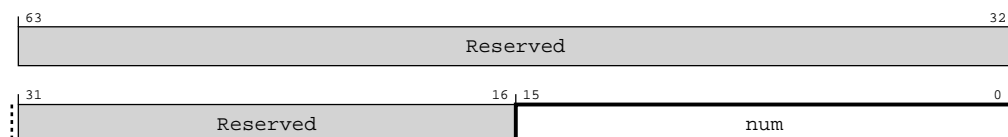


Table 5-195: cmn_hns_errdev attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:0]	num	Number of error records	RO	16'h2

5.2.4.48 cmn_hns_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'hEFD0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-192: cmn_hns_errpidr45

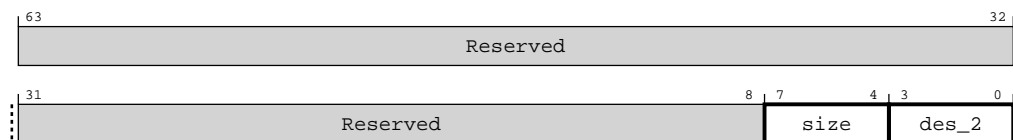


Table 5-196: cmn_hns_errpidr45 attributes

Bits	Name	Description	Type	Reset
[63:8]	Reserved	Reserved	RO	-
[7:4]	size	Size of the RAS component. 4'h0 means 4K block	RO	4'h0
[3:0]	des_2	Designer bit[10:7]	RO	4'h4

5.2.4.49 cmn_hns_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'hEFE0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-193: cmn_hns_errpidr01

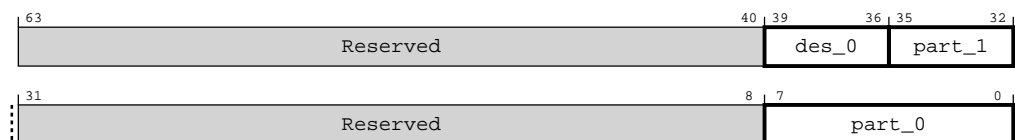


Table 5-197: cmn_hns_errpidr01 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:36]	des_0	Designer bit[3:0]	RO	4'hb
[35:32]	part_1	Product ID Part 1	RO	4'h0
[31:8]	Reserved	Reserved	RO	-
[7:0]	part_0	Product ID Part 0	RO	8'h0

5.2.4.50 cmn_hns_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'hEFE8

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-194: cmn_hns_errpidr23

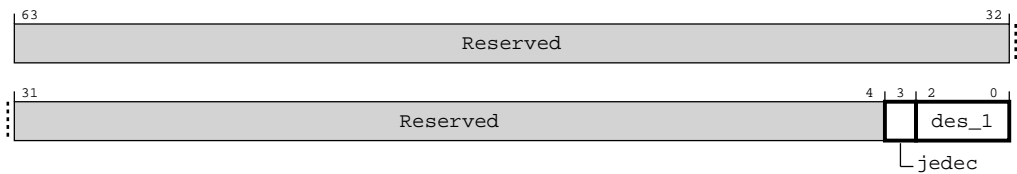


Table 5-198: cmn_hns_errpidr23 attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	jedec	JEDEC-assigned JEP106 implementer code is used.	RO	1'b1
[2:0]	des_1	Designer bit[6:4]	RO	3'h3

5.2.4.51 cmn_hns_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'hEFFF0

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-195: cmn_hns_errcidr01

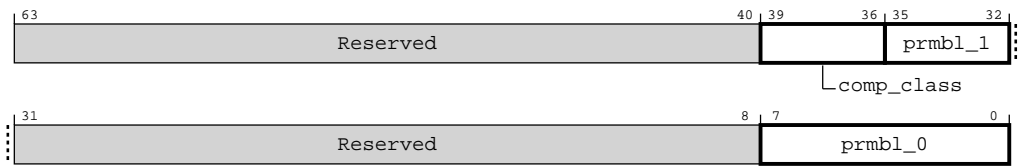


Table 5-199: cmn_hns_errcidr01 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:36]	comp_class	Component Class	RO	4'hF
[35:32]	prmb1_1	PRMBL_1	RO	4'h0
[31:8]	Reserved	Reserved	RO	-
[7:0]	prmb1_0	PRMBL_0	RO	8'hD

5.2.4.52 cmn_hns_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'hEFF8

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-196: cmn_hns_errcidr23

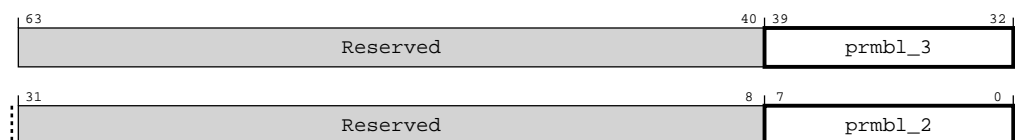


Table 5-200: cmn_hns_errcidr23 attributes

Bits	Name	Description	Type	Reset
[63:40]	Reserved	Reserved	RO	-
[39:32]	prmb1_3	PRMBL_3	RO	8'hB1
[31:8]	Reserved	Reserved	RO	-
[7:0]	prmb1_2	PRMBL_2	RO	8'h5

5.2.4.53 cmn_hns_err_inj

Enables error injection and setup. When enabled for a given source ID and logic processor ID, HN-F returns a subordinate error and reports an error interrupt. This error interrupt emulates a SLC double-bit data ECC error. This feature enables software to test the error handler. The subordinate

error is reported for cacheable read access for which SLC hit is the data source. No subordinate error or error interrupt is reported for cacheable read access in which SLC miss is the data source.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hE030

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-197: cmn_hns_err_inj

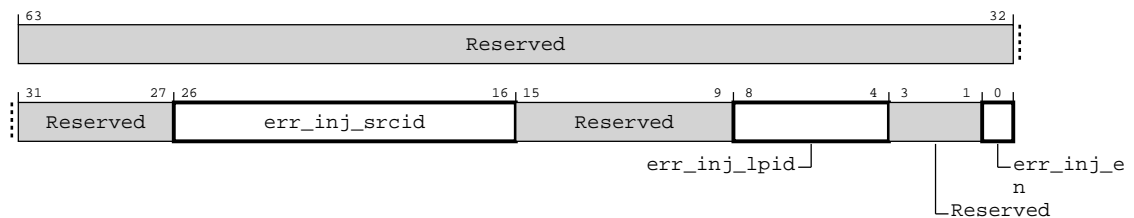


Table 5-201: cmn_hns_err_inj attributes

Bits	Name	Description	Type	Reset
[63:27]	Reserved	Reserved	RO	-
[26:16]	err_inj_srcid	RN source ID for read access which results in a SLC miss; does not report subordinate error or error to match error injection	RW	11'h0
[15:9]	Reserved	Reserved	RO	-
[8:4]	err_inj_lpid	LPID used to match for error injection	RW	5'h0
[3:1]	Reserved	Reserved	RO	-
[0]	err_inj_en	Enables error injection and report	RW	1'b0

5.2.4.54 cmn_hns_byte_par_err_inj

Functions as the byte parity error injection register for HN-F.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hE938

Type

WO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-198: cmn_hns_byte_par_err_inj

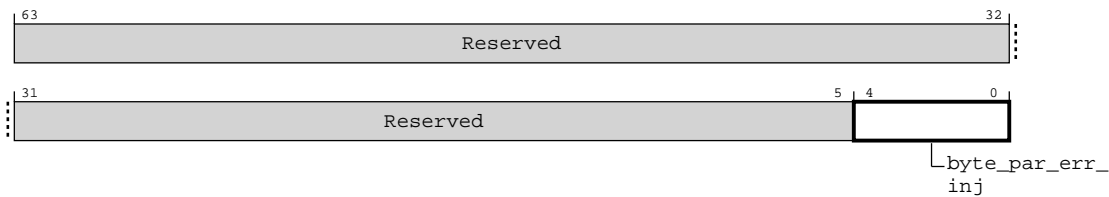


Table 5-202: cmn_hns_byte_par_err_inj attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:0]	byte_par_err_inj	Specifies a byte lane; once this register is written, a byte parity error is injected in the specified byte lane on the next SLC hit; the error will be injected in all data flits on specified byte (0 to 31)	WO	5'h0

5.2.4.55 cmn_hns_slc_lock_ways

Controls SLC way lock settings.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC00

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-199: cmn_hns_slc_lock_ways

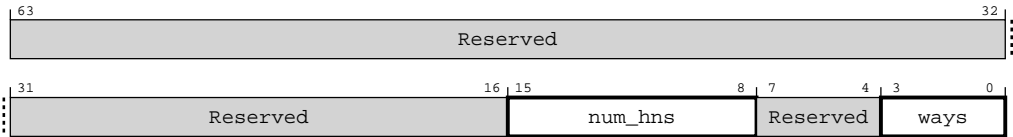


Table 5-203: cmn_hns_slc_lock_ways attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	num_hns	Number of HN-Fs in NUMA (non-uniform memory access) region	RW	Configuration dependent
[7:4]	Reserved	Reserved	RO	-
[3:0]	ways	Number of SLC ways locked (1, 2, 4, 8, 12)	RW	4'b0

5.2.4.56 cmn_hns_slc_lock_base0

Functions as the base register for lock region 0 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC08

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-200: cmn_hns_slc_lock_base0



Table 5-204: cmn_hns_slc_lock_base0 attributes

Bits	Name	Description	Type	Reset
[63]	base0_vld	Lock region 0 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base0	Lock region 0 base address	RW	52'b0

5.2.4.57 cmn_hns_slc_lock_base1

Functions as the base register for lock region 1 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC10

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.slc_lock_ways`

Secure group override

`cmn_hns_scr.slc_lock_ways`

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.slc_lock_ways` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.slc_lock_ways` bit and `cmn_hns_scr.slc_lock_ways` bit are set, Non-secure and Realm accesses to this register are permitted.

The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-201: cmn_hns_slc_lock_base1

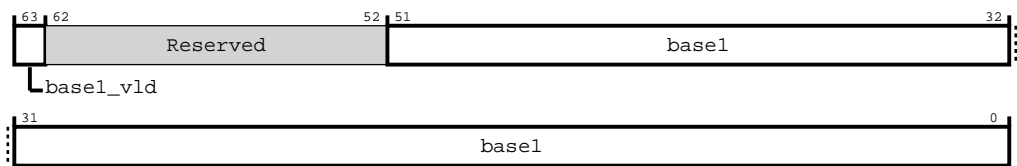


Table 5-205: cmn_hns_slc_lock_base1 attributes

Bits	Name	Description	Type	Reset
[63]	base1_vld	Lock region 1 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base1	Lock region 1 base address	RW	52'b0

5.2.4.58 cmn_hns_slc_lock_base2

Functions as the base register for lock region 2 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC18

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.slc_lock_ways` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.slc_lock_ways` bit and `cmn_hns_scr.slc_lock_ways` bit are set, Non-secure and Realm accesses to this register are permitted.

The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-202: `cmn_hns_slc_lock_base2`

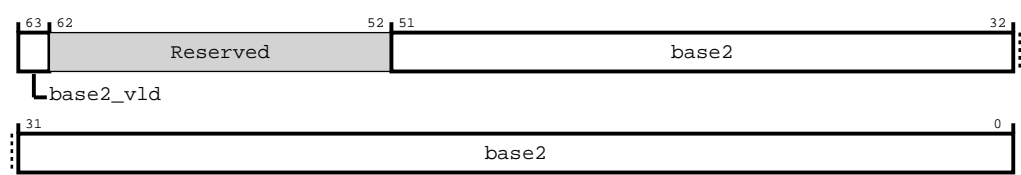


Table 5-206: `cmn_hns_slc_lock_base2` attributes

Bits	Name	Description	Type	Reset
[63]	base2_vld	Lock region 2 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base2	Lock region 2 base address	RW	52'b0

5.2.4.59 `cmn_hns_slc_lock_base3`

Functions as the base register for lock region 3 [47:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC20

Type

RW

Reset value

See individual bit resets

Root group override
cmn_hns_rcr.slc_lock_ways

Secure group override
cmn_hns_scr.slc_lock_ways

Usage constraints
This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.
If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.
The SLC must be flushed before writing to this register. Non-configuration accesses to this HN-F cannot occur before this configuration write and after the SLC flush.

Bit descriptions
The following image shows the higher register bit assignments.

Figure 5-203: cmn_hns_slc_lock_base3

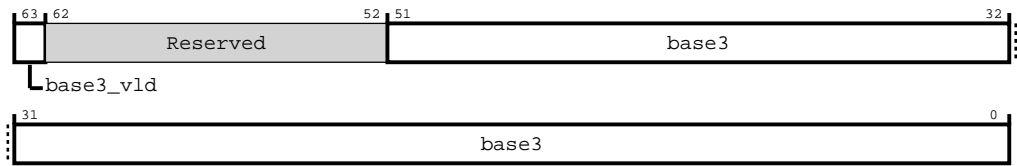


Table 5-207: cmn_hns_slc_lock_base3 attributes

Bits	Name	Description	Type	Reset
[63]	base3_vld	Lock region 3 base valid	RW	1'b0
[62:52]	Reserved	Reserved	RO	-
[51:0]	base3	Lock region 3 base address	RW	52'b0

5.2.4.60 cmn_hns_rni_region_vec

Functions as the control register for RN-I source SLC way allocation.

Configurations
This register is available in all configurations.

Attributes

Width
64

Address offset
16'hC28

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-204: cmn_hns_rni_region_vec

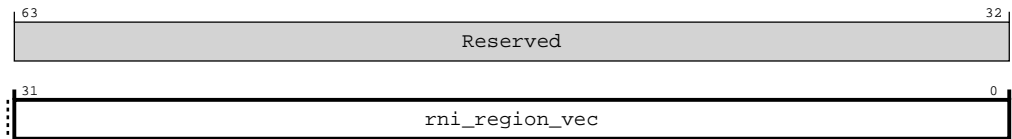


Table 5-208: cmn_hns_rni_region_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_region_vec	Bit vector mask; identifies which logical IDs of the RN-Is to allocate to the locked region NOTE: Must be set to 32'b0 if range-based region locking or OCM is enabled.	RW	32'b0

5.2.4.61 cmn_hns_rnd_region_vec

Functions as the control register for RN-D source SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC30

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-205: cmn_hns_rnd_region_vec

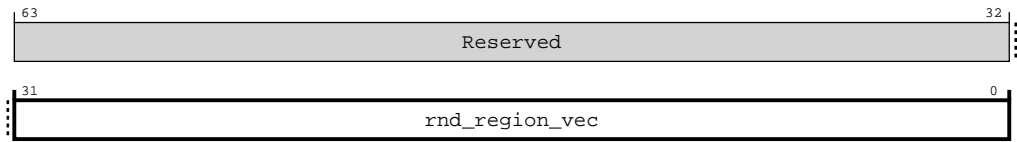


Table 5-209: cmn_hns_rnd_region_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_region_vec	Bit vector mask; identifies which logical IDs of the RN-Ds to allocate to the locked region NOTE: Must be set to 32'b0 if range-based region locking or OCM is enabled.	RW	32'b0

5.2.4.62 cmn_hns_rnf_region_vec

Functions as the control register for RN-F source SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC38

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-206: cmn_hns_rnf_region_vec

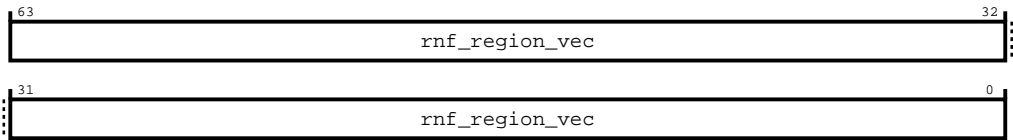


Table 5-210: cmn_hns_rnf_region_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_region_vec	Bit vector mask; identifies which logical IDs of the RN-Fs to allocate to the locked region NOTE: Must be 64'b0 if range-based region locking or OCM is enabled.	RW	64'b0

5.2.4.63 cmn_hns_rnf_region_vec1

Functions as the control register for RN-F source SLC way allocation for logical IDs 64 through 127.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hC40

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.
If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-207: cmn_hns_rnf_region_vec1

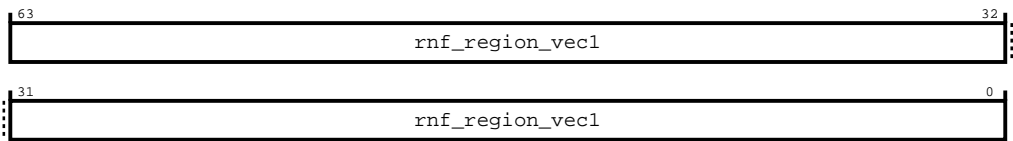


Table 5-211: cmn_hns_rnf_region_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_region_vec1	Bit vector mask; identifies which logical IDs of the RN-Fs to allocate to the locked region NOTE: Must be 64'b0 if range-based region locking or OCM is enabled.	RW	64'b0

5.2.4.64 cmn_hns_slcway_partition0_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC48

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-208: cmn_hns_slcway_partition0_rnf_vec

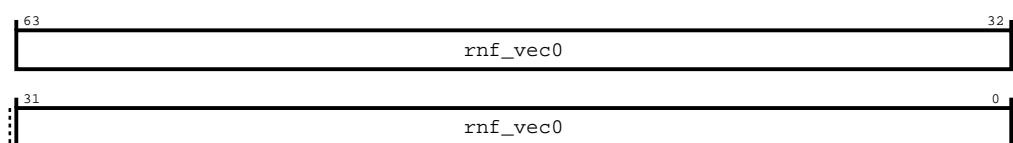


Table 5-212: cmn_hns_slcway_partition0_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec0	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.65 cmn_hns_slcway_partition1_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 1 (ways 4, 5, 6, and 7).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC50

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-209: cmn_hns_slcway_partition1_rnf_vec

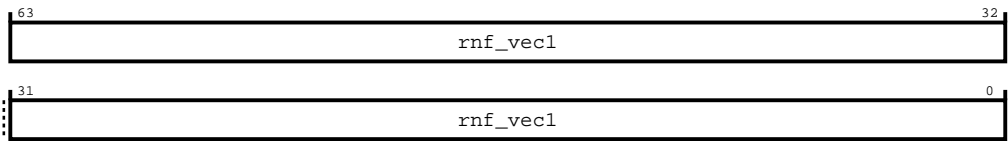


Table 5-213: cmn_hns_slcway_partition1_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec1	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.66 cmn_hns_slcway_partition2_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 2 (ways 8, 9, 10, and 11).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC58

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.
If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-210: cmn_hns_slcway_partition2_rnf_vec

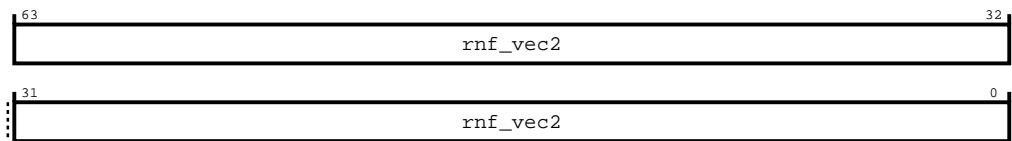


Table 5-214: cmn_hns_slcway_partition2_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec2	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.67 cmn_hns_slcway_partition3_rnf_vec

Functions as the control register for RN-Fs that can allocate to partition 3 (ways 12, 13, 14, and 15).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC60

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-211: cmn_hns_slcway_partition3_rnf_vec

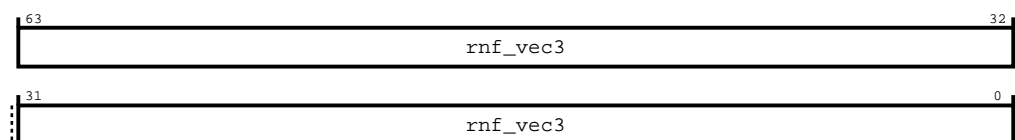


Table 5-215: cmn_hns_slcway_partition3_rnf_vec attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec3	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.68 cmn_hns_slcway_partition0_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hCB0

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-212: cmn_hns_slcway_partition0_rnf_vec1

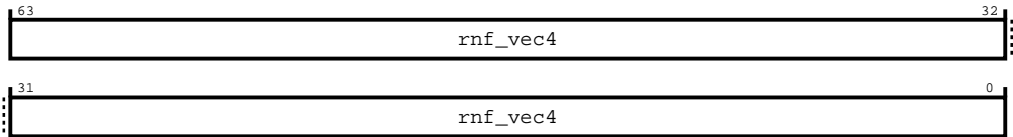


Table 5-216: cmn_hns_slcway_partition0_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec4	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.69 cmn_hns_slcway_partition1_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 1 (ways 4, 5, 6, and 7) for Logical RNF IDs 64 to 127.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hCB8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-213: cmn_hns_slcway_partition1_rnf_vec1

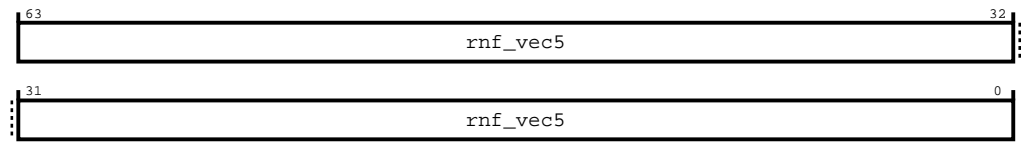


Table 5-217: cmn_hns_slcway_partition1_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec5	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.70 cmn_hns_slcway_partition2_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 2 (ways 8, 9, 10, and 11) for Logical RNF IDs 64 to 127.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hCC0

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-214: cmn_hns_slcway_partition2_rnf_vec1

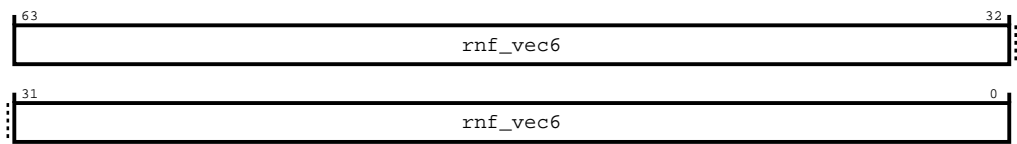


Table 5-218: cmn_hns_slcway_partition2_rnf_vec1 attributes

Bits	Name	Description	Type	Reset
[63:0]	rnf_vec6	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.71 cmn_hns_slcway_partition3_rnf_vec1

Functions as the control register for RN-Fs that can allocate to partition 3 (ways 12, 13, 14, and 15) for Logical RNF IDs 64 to 127.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hCC8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.slc_lock_ways` bit and `cmn_hns_scr.slc_lock_ways` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-215: `cmn_hns_slcway_partition3_rnf_vec1`

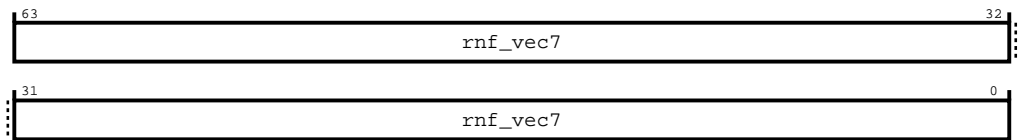


Table 5-219: `cmn_hns_slcway_partition3_rnf_vec1` attributes

Bits	Name	Description	Type	Reset
[63:0]	<code>rnf_vec7</code>	Bit vector mask; identifies which logical IDs of the RN-F can allocate	RW	64'hFFFFFFFFFFFFFFFF

5.2.4.72 `cmn_hns_slcway_partition0_rni_vec`

Functions as the control register for RN-Is that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC68

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.slc_lock_ways`

Secure group override

`cmn_hns_scr.slc_lock_ways`

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the `cmn_hns_rcr.slc_lock_ways` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.slc_lock_ways` bit and `cmn_hns_scr.slc_lock_ways` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-216: `cmn_hns_slcway_partition0_rni_vec`

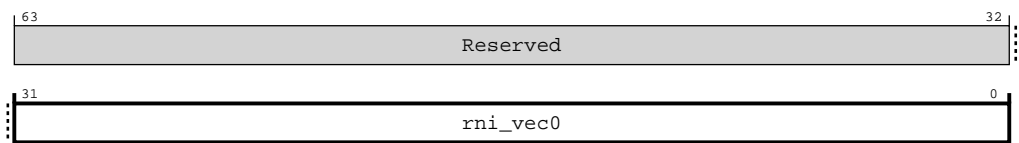


Table 5-220: `cmn_hns_slcway_partition0_rni_vec` attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec0	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.73 `cmn_hns_slcway_partition1_rni_vec`

Functions as the control register for RN-Is that can allocate to partition 1 (ways 4, 5, 6, and 7).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC70

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.slc_lock_ways`

Secure group override

`cmn_hns_scr.slc_lock_ways`

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.slc_lock_ways` bit is set, Secure accesses to this register are permitted.
If both the `cmn_hns_rcr.slc_lock_ways` bit and `cmn_hns_scr.slc_lock_ways` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-217: `cmn_hns_slcway_partition1_rni_vec`

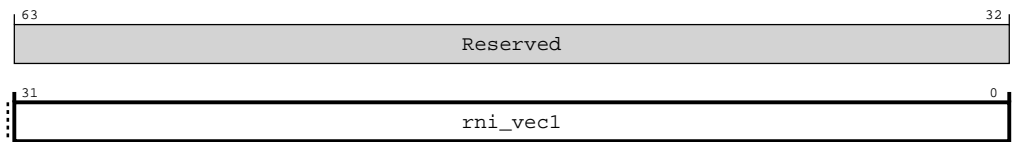


Table 5-221: `cmn_hns_slcway_partition1_rni_vec` attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec1	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.74 `cmn_hns_slcway_partition2_rni_vec`

Functions as the control register for RN-Is that can allocate to partition 2 (ways 8, 9, 10, and 11).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC78

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.slc_lock_ways`

Secure group override

`cmn_hns_scr.slc_lock_ways`

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.slc_lock_ways` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.slc_lock_ways` bit and `cmn_hns_scr.slc_lock_ways` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-218: `cmn_hns_slcway_partition2_rni_vec`

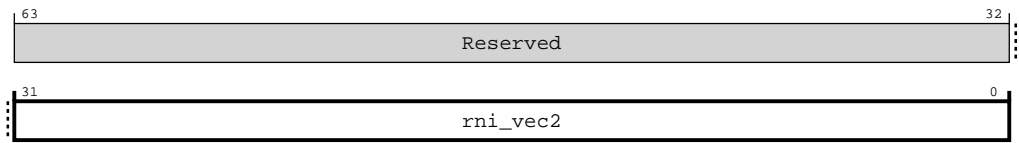


Table 5-222: `cmn_hns_slcway_partition2_rni_vec` attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec2	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.75 `cmn_hns_slcway_partition3_rni_vec`

Functions as the control register for RN-Is that can allocate to partition 3 (ways 12, 13, 14, and 15).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC80

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.slc_lock_ways`

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-219: cmn_hns_slcway_partition3_rni_vec

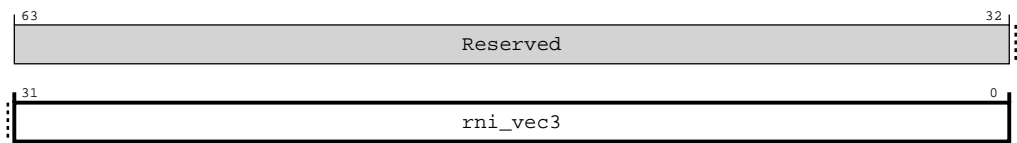


Table 5-223: cmn_hns_slcway_partition3_rni_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rni_vec3	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFF

5.2.4.76 cmn_hns_slcway_partition0_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 0 (ways 0, 1, 2, and 3).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC88

Type

RW

Reset value

See individual bit resets

Root group override
cmn_hns_rcr.slc_lock_ways

Secure group override
cmn_hns_scr.slc_lock_ways

Usage constraints
This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.
If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions
The following image shows the higher register bit assignments.

Figure 5-220: cmn_hns_slcway_partition0_rnd_vec

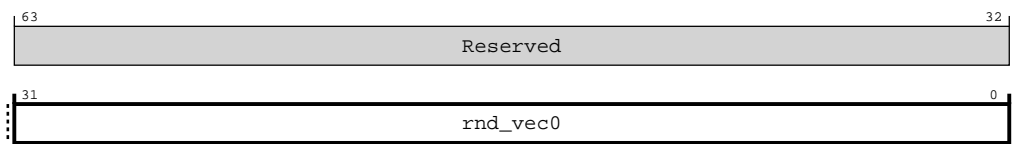


Table 5-224: cmn_hns_slcway_partition0_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec0	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.77 cmn_hns_slcway_partition1_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 1 (ways 4, 5, 6, and 7).

Configurations
This register is available in all configurations.

Attributes

Width
64

Address offset
16'hC90

Type
RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-221: cmn_hns_slcway_partition1_rnd_vec

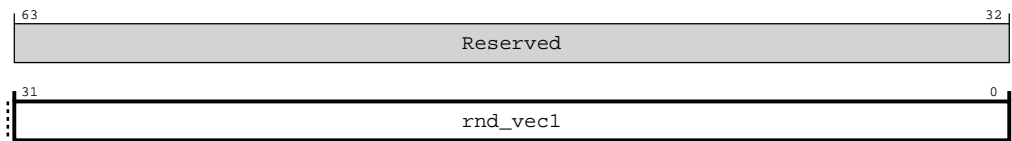


Table 5-225: cmn_hns_slcway_partition1_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec1	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.78 cmn_hns_slcway_partition2_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 2 (ways 8, 9, 10, and 11).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hC98

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-222: cmn_hns_slcway_partition2_rnd_vec

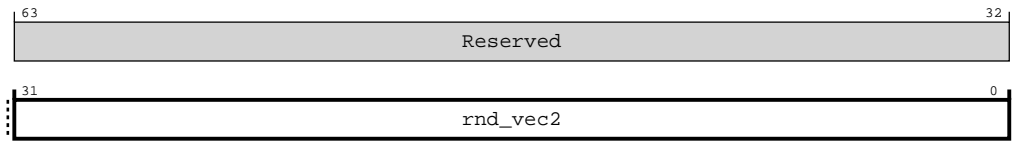


Table 5-226: cmn_hns_slcway_partition2_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec2	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.79 cmn_hns_slcway_partition3_rnd_vec

Functions as the control register for RN-Ds that can allocate to partition 3 (ways 12, 13, 14, and 15).

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCA0

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-223: cmn_hns_slcway_partition3_rnd_vec

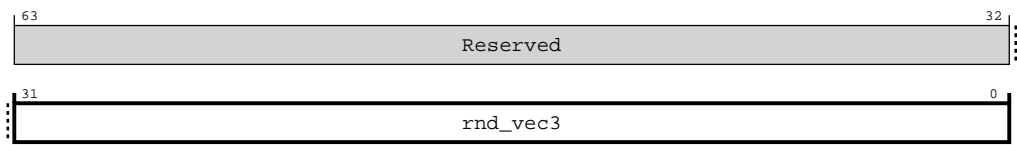


Table 5-227: cmn_hns_slcway_partition3_rnd_vec attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	rnd_vec3	Bit vector mask; identifies which logical IDs of the RN-I/RN-D can allocate	RW	32'hFFFFFFFF

5.2.4.80 cmn_hns_rn_region_lock

Functions as the enable register for source-based SLC way allocation.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCA8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slc_lock_ways

Secure group override

cmn_hns_scr.slc_lock_ways

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.slc_lock_ways bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.slc_lock_ways bit and cmn_hns_scr.slc_lock_ways bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-224: cmn_hns_rn_region_lock

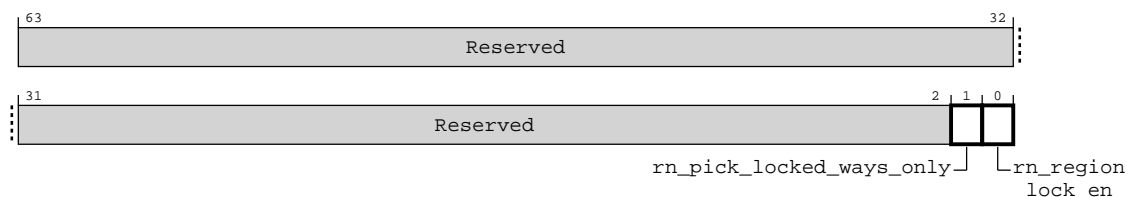


Table 5-228: cmn_hns_rn_region_lock attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1]	rn_pick_locked_ways_only	Specifies which ways the programmed RNs can allocate new cache lines to 1'b0: Programmed RN will choose all ways including locked 1'b1: Programmed RN will only allocate in locked ways	RW	1'b0
[0]	rn_region_lock_en	Enables SRC-based region locking 1'b0: SRC based way locking is disabled 1'b1: SRC based way locking is enabled	RW	1'b0

5.2.4.81 cmn_hns_sf_cxg_blocked_ways

Specifies the SF ways that are blocked for remote chip to use in CML mode.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCD0

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-225: cmn_hns_sf_cxg_blocked_ways

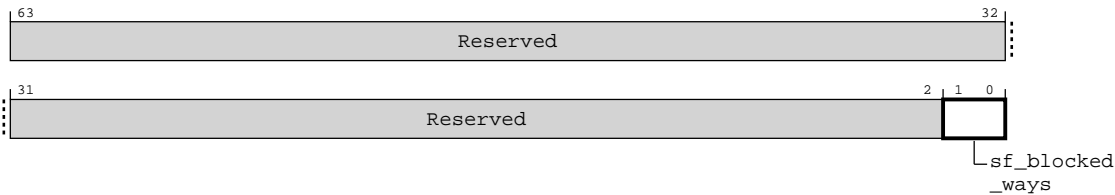


Table 5-229: cmn_hns_sf_cxg_blocked_ways attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[1:0]	sf_blocked_ways	Number of SF ways blocked from remote chips to be able to use in CML mode. 2'b00: No ways are blocked; all SF ways could be used by local or remote RN-Fs 2'b01: SF_NUM_WAYS = 16: ways 3:0 for local RN-Fs only; ways 15:4 for local and remote RN-Fs SF_NUM_WAYS > 16: ways 7:0 for local RN-Fs only; ways 31:8 for local and remote RN-Fs 2'b10: SF_NUM_WAYS = 16: ways 7:0 for local RN-Fs only; ways 15:8 for local and remote RN-Fs SF_NUM_WAYS > 16: ways 15:0 for local RN-Fs only; ways 31:16 for local and remote RN-Fs 2'b11: SF_NUM_WAYS < 26: ways 11:0 for local RN-Fs only; ways 15:12 for local and remote RN-Fs SF_NUM_WAYS >= 26: ways 23:0 for local RN-Fs only; ways 31:24 for local and remote RN-Fs	RW	2'b00

5.2.4.82 cmn_hns_cxg_ha_metadata_exclusion_list

Functions as the control register to identify CXG HA which does not support metadata

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hCE0

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-226: cmn_hns_cxg_ha_metadata_exclusion_list

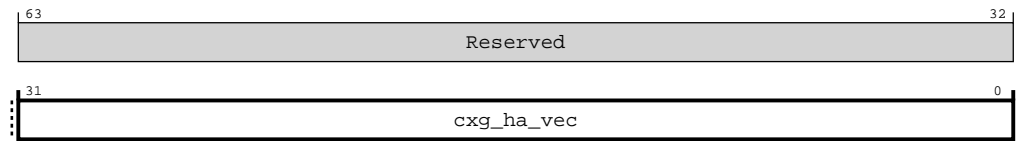


Table 5-230: cmn_hns_cxg_ha_metadata_exclusion_list attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	cxg_ha_vec	Bit vector mask; identifies which logical IDs of the CXG HA does not support metadata	RW	32'h00000000

5.2.4.83 cmn_hns_cxg_ha_smp_exclusion_list

Functions as the control register to identify CXG HA not connected to SMP CCIX link

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCD8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.sam_control` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.sam_control` bit and `cmn_hns_scr.sam_control` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-227: cmn_hns_cxg_ha_smp_exclusion_list

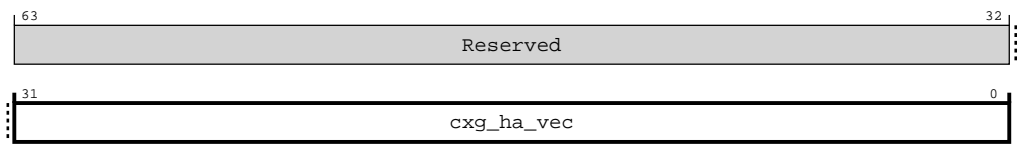


Table 5-231: cmn_hns_cxg_ha_smp_exclusion_list attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:0]	cxg_ha_vec	Bit vector mask; identifies which logical IDs of the CXG HA does not connect to SMP CCIX link	RW	32'h00000000

5.2.4.84 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

This register is owned in the Root space and accessible using Root transactions.
Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-228: hn_sam_hash_addr_mask_reg

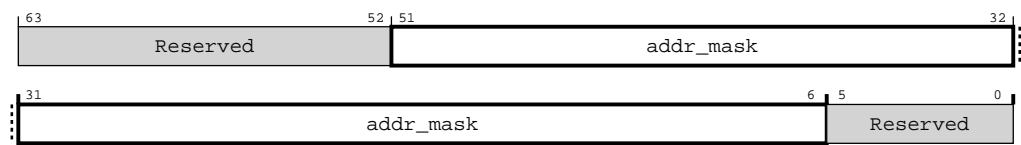


Table 5-232: hn_sam_hash_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address mask applied before hashing	RW	46'h3FFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

5.2.4.85 hn_sam_region_cmp_addr_mask_reg

Configures the address mask that is applied before memory region compare.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hCF8

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.
Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-229: hn_sam_region_cmp_addr_mask_reg

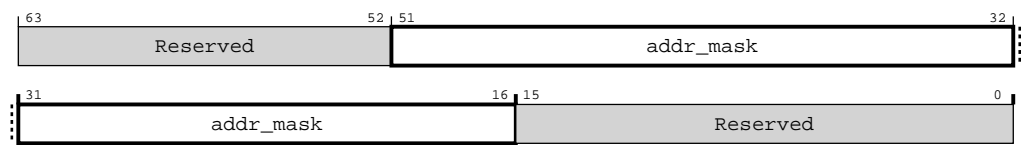


Table 5-233: hn_sam_region_cmp_addr_mask_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	addr_mask	Address mask applied before memory region compare	RW	36'hFFFFFFFF
[15:0]	Reserved	Reserved	RO	-

5.2.4.86 cmn_hns_sam_cfg1_def_hashed_region

Configures default hashed region in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD48

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-230: cmn_hns_sam_cfg1_def_hashed_region

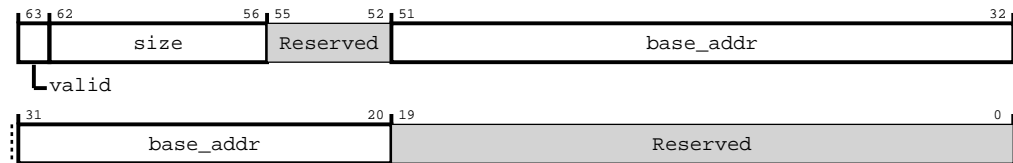


Table 5-234: cmn_hns_sam_cfg1_def_hashed_region attributes

Bits	Name	Description	Type	Reset
[63]	valid	Default hashed region valid 1'b0: not valid 1'b1: valid for memory region comparison	RW	1'h1
[62:56]	size	Default hashed region 0 size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size ($2^{\text{address width}}$).	RW	7'h7F
[55:52]	Reserved	Reserved	RO	-
[51:20]	base_addr	Bits [51:20] of base address of the range, LSB bit is defined by the parameter <code>POR_HNSAM_RCOMP_LSB_PARAM</code>	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

5.2.4.87 cmn_hns_sam_cfg2_def_hashed_region

Configures default hashed region in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD50

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.sam_control`

Secure group override

`cmn_hns_scr.sam_control`

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.sam_control` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.sam_control` bit and `cmn_hns_scr.sam_control` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-231: `cmn_hns_sam_cfg2_def_hashed_region`

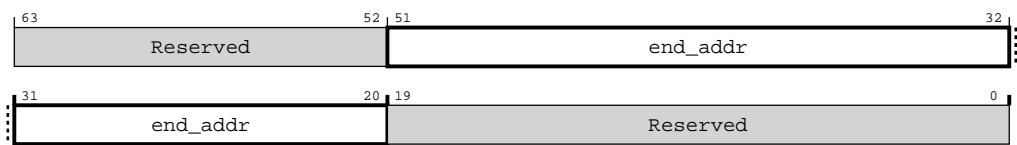


Table 5-235: `cmn_hns_sam_cfg2_def_hashed_region` attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	end_addr	Bits [51:20] of end address of the range, LSB bit is defined by the parameter <code>POR_HNSAM_RCOMP_LSB_PARAM</code>	RW	32'hFFFFFFFF
[19:0]	Reserved	Reserved	RO	-

5.2.4.88 `cmn_hns_sam_control`

Configures HN-F SAM. All `top_address_bit` fields must be between bits 47 and 28 of the address. `top_address_bit2 > top_address_bit1 > top_address_bit0`. Must be configured to match corresponding `por_rnsam_sys_cache_grp_sn_sam_cfgN` register in the RN SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD00

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-232: cmn_hns_sam_control

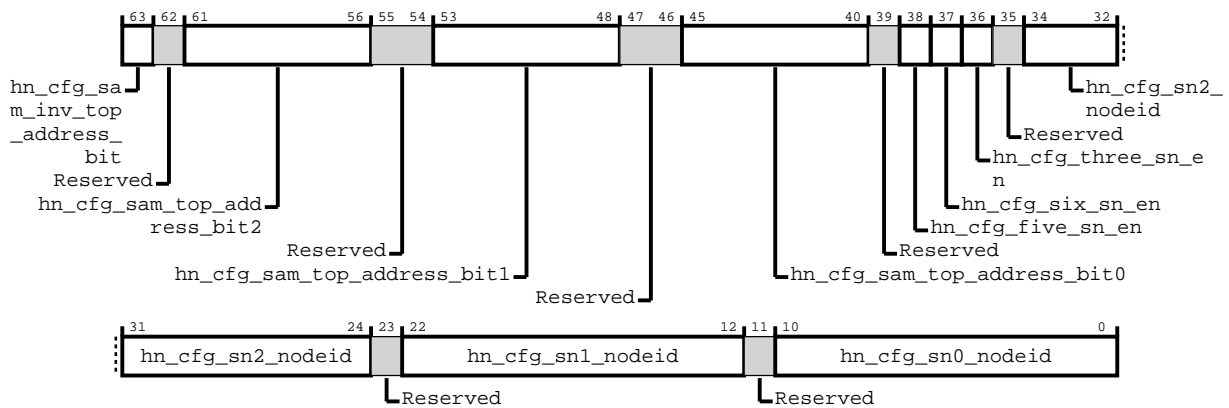


Table 5-236: cmn_hns_sam_control attributes

Bits	Name	Description	Type	Reset
[63]	hn_cfg_sam_inv_top_address_bit	Inverts the top address bit (hn_cfg_sam_top_address_bit1 if 3-SN, hn_cfg_sam_top_address_bit2 if 6-SN) NOTE: Can only be used when the address map does not have unique address bit combinations.	RW	1'h0
[62]	Reserved	Reserved	RO	-
[61:56]	hn_cfg_sam_top_address_bit2	Bit position of top_address_bit2; used for address hashing in 6-SN configuration	RW	6'h00
[55:54]	Reserved	Reserved	RO	-
[53:48]	hn_cfg_sam_top_address_bit1	Bit position of top_address_bit1; used for address hashing in 3-SN/6-SN configuration	RW	6'h00
[47:46]	Reserved	Reserved	RO	-
[45:40]	hn_cfg_sam_top_address_bit0	Bit position of top_address_bit0; used for address hashing in 3-SN/6-SN configuration	RW	6'h00

Bits	Name	Description	Type	Reset
[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

5.2.4.89 cmn_hns_sam_control2

Configures HN-F SAM. Must be configured to match corresponding por_rnsam_sys_cache_grp_sn_sam_cfgN register in the RN SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD28

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-233: cmn_hns_sam_control2

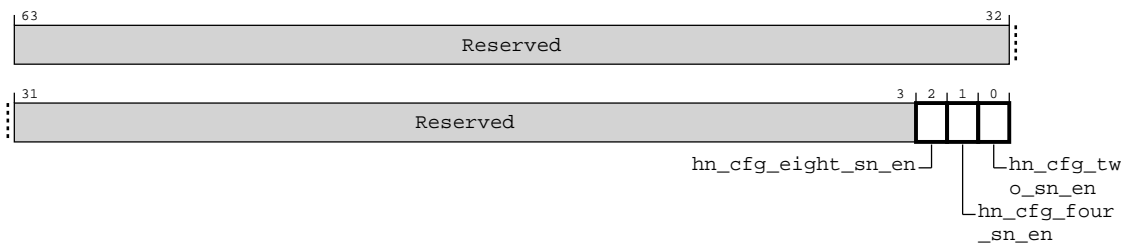


Table 5-237: cmn_hns_sam_control2 attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2]	hn_cfg_eight_sn_en	Enables 8-SN configuration	RW	1'b0
[1]	hn_cfg_four_sn_en	Enables 4-SN configuration	RW	1'b0
[0]	hn_cfg_two_sn_en	Enables 2-SN configuration	RW	1'b0

5.2.4.90 cmn_hns_sam_memregion0

Configures range-based memory region 0 in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD08

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.sam_control` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.sam_control` bit and `cmn_hns_scr.sam_control` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-234: `cmn_hns_sam_memregion0`

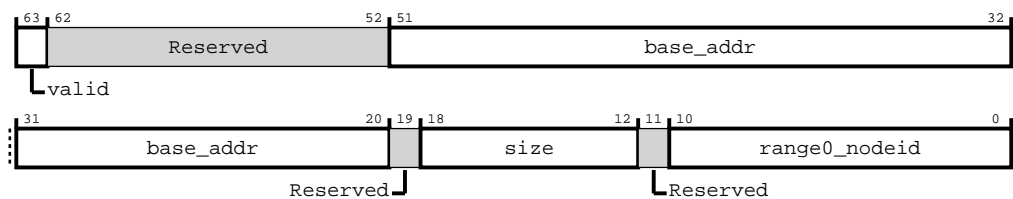


Table 5-238: `cmn_hns_sam_memregion0` attributes

Bits	Name	Description	Type	Reset
[63]	valid	Memory region 0 valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'h0
[62:52]	Reserved	Reserved	RO	-
[51:20]	base_addr	Base address of memory region 0 CONSTRAINT: Must be an integer multiple of region size.	RW	32'h0
[19]	Reserved	Reserved	RO	-
[18:12]	size	Memory region 0 size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'h0
[11]	Reserved	Reserved	RO	-
[10:0]	range0_nodeid	Memory region 0 target node ID	RW	11'h0

5.2.4.91 `cmn_hns_sam_memregion0_end_addr`

Configures end address memory region 0 in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD38

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-235: cmn_hns_sam_memregion0_end_addr

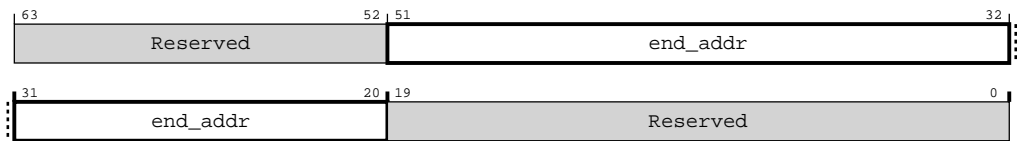


Table 5-239: cmn_hns_sam_memregion0_end_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	end_addr	End address of memory region 0	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

5.2.4.92 cmn_hns_sam_memregion1

Configures range-based memory region 1 in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD10

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-236: cmn_hns_sam_memregion1

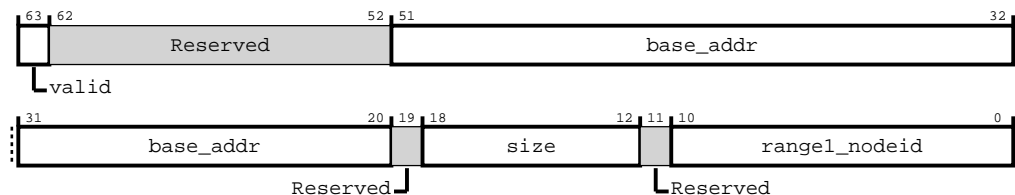


Table 5-240: cmn_hns_sam_memregion1 attributes

Bits	Name	Description	Type	Reset
[63]	valid	Memory region 1 valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'h0
[62:52]	Reserved	Reserved	RO	-
[51:20]	base_addr	Base address of memory region 1 CONSTRAINT: Must be an integer multiple of region size.	RW	32'h0
[19]	Reserved	Reserved	RO	-
[18:12]	size	Memory region 1 size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'h0
[11]	Reserved	Reserved	RO	-
[10:0]	range1_nodeid	Memory region 1 target node ID	RW	11'h0

5.2.4.93 cmn_hns_sam_memregion1_end_addr

Configures end address memory region 1 in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD40

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-237: cmn_hns_sam_memregion1_end_addr

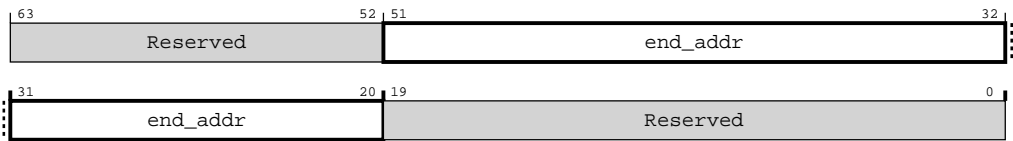


Table 5-241: cmn_hns_sam_memregion1_end_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[51:20]	end_addr	End address of memory region 1	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

5.2.4.94 cmn_hns_sam_sn_properties

Configures properties for all six SN targets and two range-based SN targets.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD18

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-238: cmn_hns_sam_sn_properties

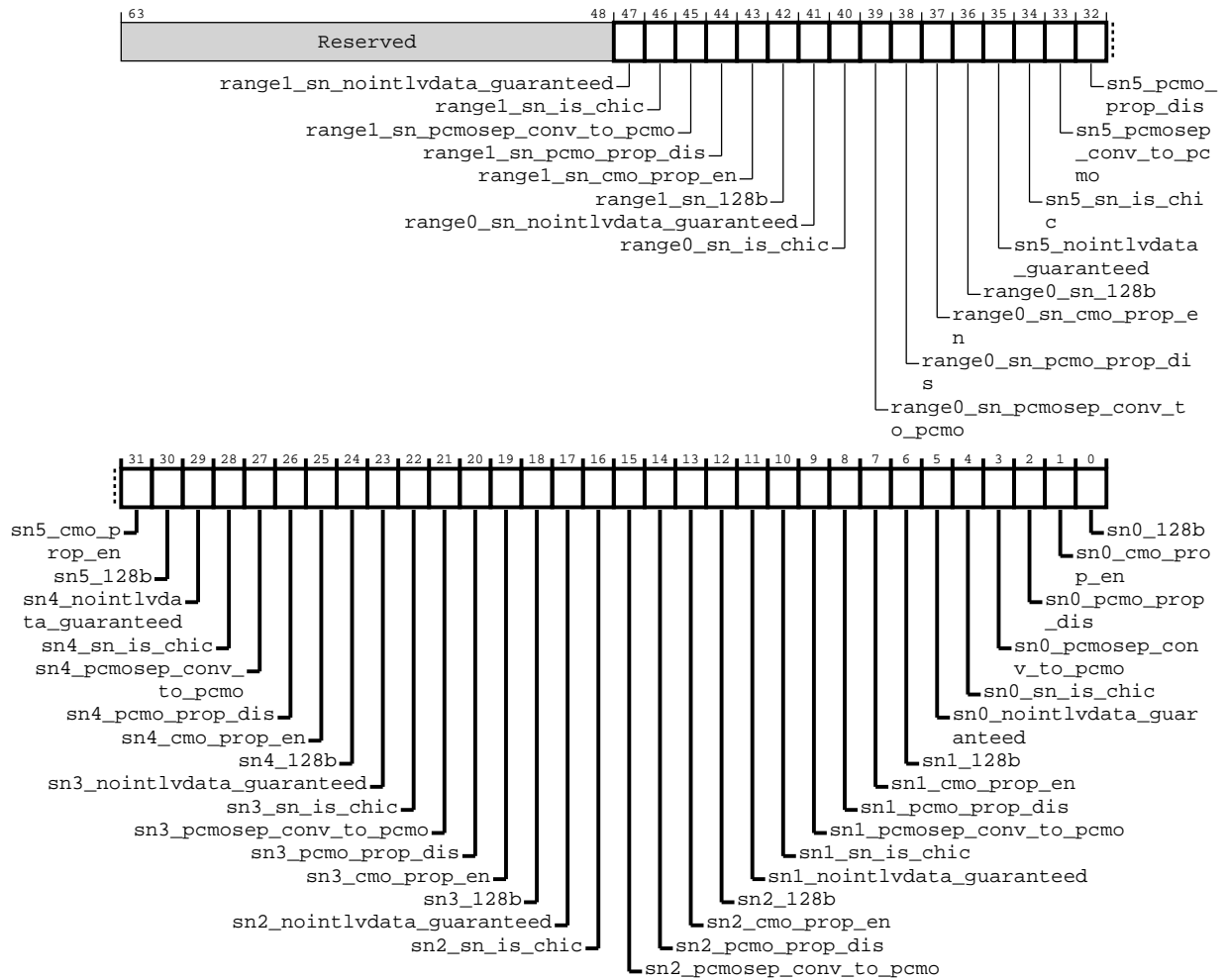


Table 5-242: cmn_hns_sam_sn_properties attributes

Bits	Name	Description	Type	Reset
[63:48]	Reserved	Reserved	RO	-
[47]	range1_sn_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[46]	range1_sn_is_chic	Indicates that the range 1 SN is a CHI-C SN when set	RW	1'b0
[45]	range1_sn_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for range 1 SN when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[44]	range1_sn_pcmo_prop_dis	Disables PCMO (persistent CMO) propagation for range 1 SN when set	RW	1'b0
[43]	range1_sn_cmo_prop_en	Enables CMO propagation for range 1 SN	RW	1'b0
[42]	range1_sn_128b	Data width of range 1 SN 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[41]	range0_sn_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[40]	range0_sn_is_chic	Indicates that the range 0 SN is a CHI-C SN when set	RW	1'b0

Bits	Name	Description	Type	Reset
[39]	range0_sn_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for range 0 SN when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[38]	range0_sn_pcmo_prop_dis	Disables PCMO (persistent CMO) propagation for range 0 SN when set	RW	1'b0
[37]	range0_sn_cmo_prop_en	Enables CMO propagation for range 0 SN	RW	1'b0
[36]	range0_sn_128b	Data width of range 0 SN 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[35]	sn5_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[34]	sn5_sn_is_chic	Indicates that SN5 is a CHI-C SN when set	RW	1'b0
[33]	sn5_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 5 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[32]	sn5_pcmo_prop_dis	Disables PCMO propagation for SN 5 when set	RW	1'b0
[31]	sn5_cmo_prop_en	Enables CMO propagation for SN 5 when set	RW	1'b0
[30]	sn5_128b	Data width of SN 5 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[29]	sn4_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[28]	sn4_sn_is_chic	Indicates that SN4 is a CHI-C SN when set	RW	1'b0
[27]	sn4_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 4 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[26]	sn4_pcmo_prop_dis	Disables PCMO propagation for SN 4 when set	RW	1'b0
[25]	sn4_cmo_prop_en	Enables CMO propagation for SN 4 when set	RW	1'b0
[24]	sn4_128b	Data width of SN 4 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[23]	sn3_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[22]	sn3_sn_is_chic	Indicates that SN3 is a CHI-C SN when set	RW	1'b0
[21]	sn3_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 3 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[20]	sn3_pcmo_prop_dis	Disables PCMO propagation for SN 3 when set	RW	1'b0
[19]	sn3_cmo_prop_en	Enables CMO propagation for SN 3 when set	RW	1'b0
[18]	sn3_128b	Data width of SN 3 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[17]	sn2_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[16]	sn2_sn_is_chic	Indicates that SN2 is a CHI-C SN when set	RW	1'b0
[15]	sn2_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 2 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[14]	sn2_pcmo_prop_dis	Disables PCMO propagation for SN 2 when set	RW	1'b0
[13]	sn2_cmo_prop_en	Enables CMO propagation for SN 2 when set	RW	1'b0
[12]	sn2_128b	Data width of SN 2 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[11]	sn1_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[10]	sn1_sn_is_chic	Indicates that SN1 is a CHI-C SN when set	RW	1'b0
[9]	sn1_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 1 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[8]	sn1_pcmo_prop_dis	Disables PCMO propagation for SN 1 when set	RW	1'b0
[7]	sn1_cmo_prop_en	Enables CMO propagation for SN 1 when set	RW	1'b0
[6]	sn1_128b	Data width of SN 1 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[5]	sn0_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	sn0_sn_is_chic	Indicates that SN0 is a CHI-C SN when set	RW	1'b0
[3]	sn0_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 0 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	sn0_pcmo_prop_dis	Disables PCMO propagation for SN 0 when set	RW	1'b0
[1]	sn0_cmo_prop_en	Enables CMO propagation for SN 0 when set	RW	1'b0
[0]	sn0_128b	Data width of SN 0 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0

5.2.4.95 cmn_hns_sam_6sn_nodeid

Configures node IDs for subordinate nodes 3 to 5 in 6-SN configuration mode.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD20

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-239: cmn_hns_sam_6sn_nodeid

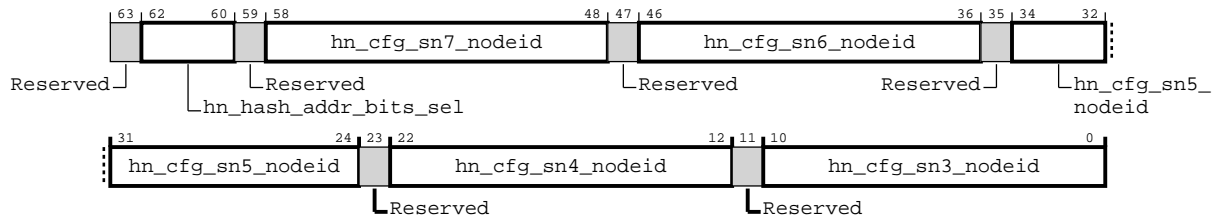


Table 5-243: cmn_hns_sam_6sn_nodeid attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:60]	hn_hash_addr_bits_sel	SN hash address select(Valid for 3SN, 5SN, 6SN) 3'b000: [16:8] address bits (Default) 3'b001: [17:9] address bits 3'b010: [18:10] address bits 3'b011: [19:11] address bits 3'b100: [20:12] address bits 3'b101: [21:13] address bits Others: Reserved	RW	3'h0
[59]	Reserved	Reserved	RO	-
[58:48]	hn_cfg_sn7_nodeid	SN 7 node ID	RW	11'h0
[47]	Reserved	Reserved	RO	-
[46:36]	hn_cfg_sn6_nodeid	SN 6 node ID	RW	11'h0
[35]	Reserved	Reserved	RO	-
[34:24]	hn_cfg_sn5_nodeid	SN 5 node ID	RW	11'h0
[23]	Reserved	Reserved	RO	-
[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

5.2.4.96 cmn_hns_sam_sn_properties1

Configures additional properties for all six SN targets and two range-based SN targets.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hCE8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-240: cmn_hns_sam_sn_properties1

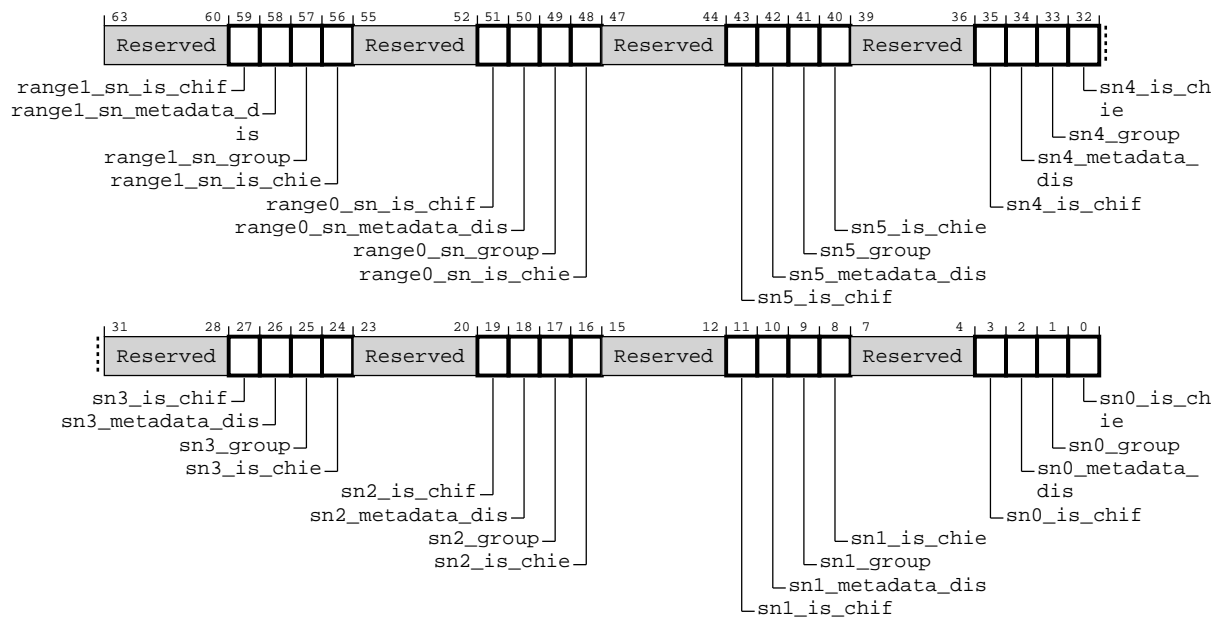


Table 5-244: cmn_hns_sam_sn_properties1 attributes

Bits	Name	Description	Type	Reset
[63:60]	Reserved	Reserved	RO	-
[59]	range1_sn_is_chif	Range 1 SN supports CHI-F	RW	1'b0
[58]	range1_sn_metadata_dis	HNS implements metadata termination flow for Range 1 SN when set	RW	1'b0
[57]	range1_sn_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[56]	range1_sn_is_chie	Range 1 SN supports CHI-E	RW	1'b0

Bits	Name	Description	Type	Reset
[55:52]	Reserved	Reserved	RO	-
[51]	range0_sn_is_chif	Range 0 SN supports CHI-F	RW	1'b0
[50]	range0_sn_metadata_dis	HNS implements metadata termination flow for Range 0 SN when set	RW	1'b0
[49]	range0_sn_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[48]	range0_sn_is_chie	Range 0 SN supports CHI-E	RW	1'b0
[47:44]	Reserved	Reserved	RO	-
[43]	sn5_is_chif	SN5 supports CHI-F	RW	1'b0
[42]	sn5_metadata_dis	HNS implements metadata termination flow for SN 5 when set	RW	1'b0
[41]	sn5_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[40]	sn5_is_chie	SN 5 supports CHI-E	RW	1'b0
[39:36]	Reserved	Reserved	RO	-
[35]	sn4_is_chif	SN4 supports CHI-F	RW	1'b0
[34]	sn4_metadata_dis	HNS implements metadata termination flow for SN 4 when set	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	RW	1'b0
[31:28]	Reserved	Reserved	RO	-
[27]	sn3_is_chif	SN3 supports CHI-F	RW	1'b0
[26]	sn3_metadata_dis	HNS implements metadata termination flow for SN 3 when set	RW	1'b0
[25]	sn3_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[24]	sn3_is_chie	SN 3 supports CHI-E	RW	1'b0
[23:20]	Reserved	Reserved	RO	-
[19]	sn2_is_chif	SN2 supports CHI-F	RW	1'b0
[18]	sn2_metadata_dis	HNS implements metadata termination flow for SN 2 when set	RW	1'b0
[17]	sn2_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[16]	sn2_is_chie	SN 2 supports CHI-E	RW	1'b0
[15:12]	Reserved	Reserved	RO	-
[11]	sn1_is_chif	SN1 supports CHI-F	RW	1'b0
[10]	sn1_metadata_dis	HNS implements metadata termination flow for SN 1 when set	RW	1'b0
[9]	sn1_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[8]	sn1_is_chie	SN 1 supports CHI-E	RW	1'b0
[7:4]	Reserved	Reserved	RO	-
[3]	sn0_is_chif	SN0 supports CHI-F	RW	1'b0
[2]	sn0_metadata_dis	HNS implements metadata termination flow for SN 0 when set	RW	1'b0
[1]	sn0_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[0]	sn0_is_chie	SN 0 supports CHI-E	RW	1'b0

5.2.4.97 cmn_hns_sam_sn_properties2

Configures properties for SN-7 & SN-8.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD30

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-241: cmn_hns_sn_properties2

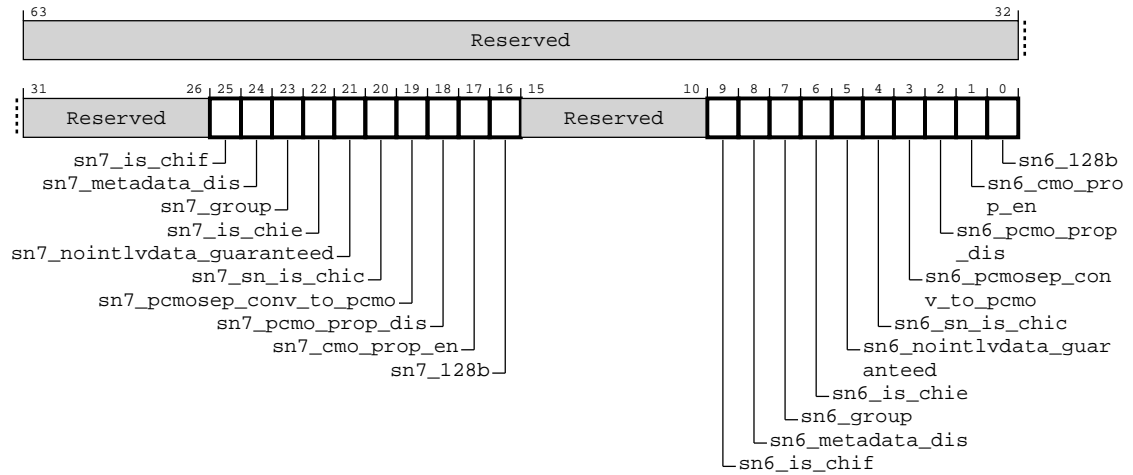


Table 5-245: cmn_hns_sn_properties2 attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25]	sn7_is_chif	SN7 supports CHI-F	RW	1'b0
[24]	sn7_metadata_dis	HNS implements metadata termination flow for SN 7 when set	RW	1'b0
[23]	sn7_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[22]	sn7_is_chie	SN 7 supports CHI-E	RW	1'b0
[21]	sn7_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[20]	sn7_sn_is_chic	Indicates that SN7 is a CHI-C SN when set	RW	1'b0
[19]	sn7_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 7 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[18]	sn7_pcmo_prop_dis	Disables PCMO propagation for SN 7 when set	RW	1'b0
[17]	sn7_cmo_prop_en	Enables CMO propagation for SN 7 when set	RW	1'b0
[16]	sn7_128b	Data width of SN 7 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[15:10]	Reserved	Reserved	RO	-
[9]	sn6_is_chif	SN6 supports CHI-F	RW	1'b0
[8]	sn6_metadata_dis	HNS implements metadata termination flow for SN 6 when set	RW	1'b0
[7]	sn6_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[6]	sn6_is_chie	SN 6 supports CHI-E	RW	1'b0
[5]	sn6_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[4]	sn6_sn_is_chic	Indicates that SN6 is a CHI-C SN when set	RW	1'b0
[3]	sn6_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN 6 when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	sn6_pcmo_prop_dis	Disables PCMO propagation for SN 6 when set	RW	1'b0
[1]	sn6_cmo_prop_en	Enables CMO propagation for SN 6 when set	RW	1'b0
[0]	sn6_128b	Data width of SN 6 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0

5.2.4.98 cmn_hns_cml_port_aggr_grp0-4_add_mask

There are 5 iterations of this register. The index ranges from 0 to 4. Configures the CCIX port aggregation address mask for group 0.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

0xindex(0-4) : 16'hF80 + #{8 * index}
 0xindex(5-31) : 16'h6000 + #{8 * index}

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-242: cmn_hns_cml_port_aggr_grp0-4_add_mask

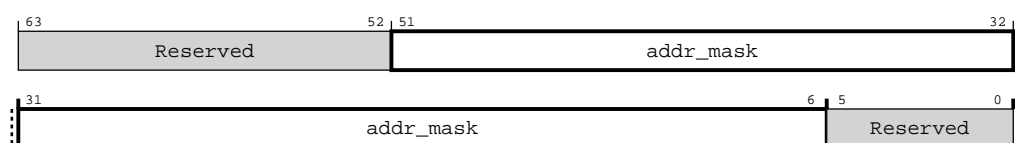


Table 5-246: cmn_hns_cml_port_aggr_grp0-4_add_mask attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask	Address mask to be applied before hashing	RW	46'h3FFFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

5.2.4.99 cmn_hns_cml_port_aggr_grp5-31_add_mask

There are 27 iterations of this register. The index ranges from 5 to 31. Configures the CCIX port aggregation address mask for group 0.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-4) : 16'hF80 + \{8 * index\}$

$0xindex(5-31) : 16'h6000 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-243: cmn_hns_cml_port_aggr_grp5-31_add_mask

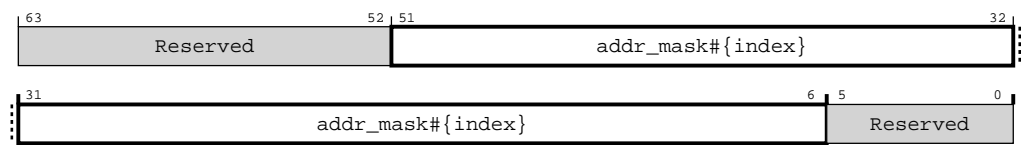


Table 5-247: cmn_hns_cml_port_aggr_grp5-31_add_mask attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:6]	addr_mask#{index}	Address mask to be applied before hashing	RW	46'h3FFFFFFFFFFFF
[5:0]	Reserved	Reserved	RO	-

5.2.4.100 cmn_hns_cml_port_aggr_grp_reg0-12

There are 13 iterations of this register. The index ranges from 0 to 12. Configures the CCIX port aggregation port Node IDs.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

0xindex(0-1) : 16'hFB0 + #{8 * index}
0xindex(2-12) : 16'h6100 + #{8 * index}

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.sam_control` bit and `cmn_hns_scr.sam_control` bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-244: `cmn_hns_cml_port_aggr_grp_reg0-12`

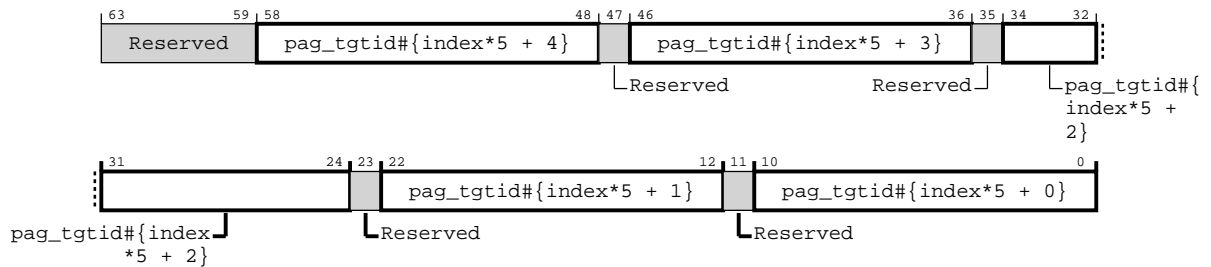


Table 5-248: `cmn_hns_cml_port_aggr_grp_reg0-12` attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	<code>pag_tgtid#{index*5 + 4}</code>	Specifies the target ID <code>#{index*5 + 4}</code> for CPAG	RW	11'b0
[47]	Reserved	Reserved	RO	-
[46:36]	<code>pag_tgtid#{index*5 + 3}</code>	Specifies the target ID <code>#{index*5 + 3}</code> for CPAG	RW	11'b0
[35]	Reserved	Reserved	RO	-
[34:24]	<code>pag_tgtid#{index*5 + 2}</code>	Specifies the target ID <code>{index*5 + 2}</code> for CPAG	RW	11'b0
[23]	Reserved	Reserved	RO	-
[22:12]	<code>pag_tgtid#{index*5 + 1}</code>	Specifies the target ID <code>{index*5 + 1}</code> for CPAG	RW	11'b0
[11]	Reserved	Reserved	RO	-
[10:0]	<code>pag_tgtid#{index*5 + 0}</code>	Specifies the target ID <code>{index*5 + 0}</code> for CPAG	RW	11'b0

5.2.4.101 `cmn_hns_cml_port_aggr_ctrl_reg`

Configures the CCIX port aggregation port groups.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hFD0

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-245: cmn_hns_cml_port_aggr_ctrl_reg

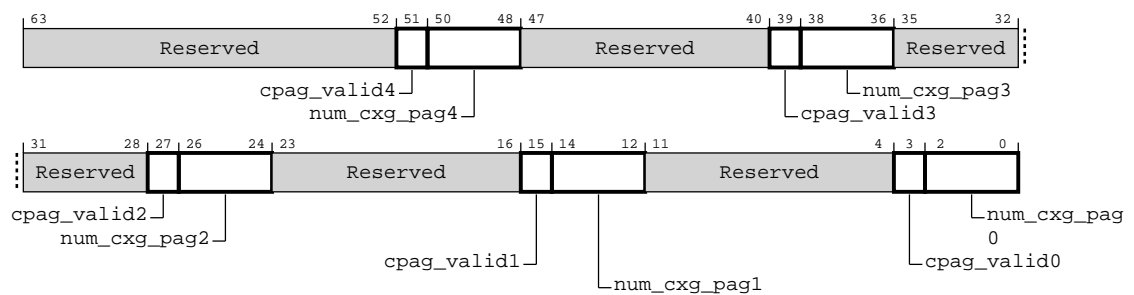


Table 5-249: cmn_hns_cml_port_aggr_ctrl_reg attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	cpag_valid4	Valid programming for CPAG4, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[50:48]	num_cxg_pag4	Specifies the number of CXRAs in CPAG4 Constraint: May use pag_tgtid8 through pag_tgtid9 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 3'b000 1 port used 3'b001 2 ports used 3'b010 4 ports used 3'b011 8 ports used 3'b100 16 ports used 3'b101 32 ports used 3'b110 3 ports (MOD-3 hash) 3'b111 Reserved	RW	3'b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid3	Valid programming for CPAG + 3}, Enabled by default (backward compatible)	RW	1'b1
[38:36]	num_cxg_pag3	Specifies the number of CXRAs in CPAG3 Constraint: May use pag_tgtid6 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 3'b000 1 port used 3'b001 2 ports used 3'b010 4 ports used 3'b011 8 ports used 3'b100 16 ports used 3'b101 32 ports used 3'b110 3 ports (MOD-3 hash) 3'b111 Reserved	RW	3'b000
[35:28]	Reserved	Reserved	RO	-
[27]	cpag_valid2	Valid programming for CPAG + 2}, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[26:24]	num_cxg_pag2	Specifies the number of CXRAs in CPAG2 Constraint: May use pag_tgtid4 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 3'b000 1 port used 3'b001 2 ports used 3'b010 4 ports used 3'b011 8 ports used 3'b100 16 ports used 3'b101 32 ports used 3'b110 3 ports (MOD-3 hash) 3'b111 Reserved	RW	3'b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid1	Valid programming for CPAG + 1}, Enabled by default (backward compatible)	RW	1'b1
[14:12]	num_cxg_pag1	Specifies the number of CXRAs in CPAG1 Constraint: May use pag_tgtid2 through pag_tgtid3 of cmn_hns_cml_port_aggr_grp_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0 3'b000 1 port used 3'b001 2 ports used 3'b010 4 ports used 3'b011 8 ports used 3'b100 16 ports used 3'b101 32 ports used 3'b110 3 ports (MOD-3 hash) 3'b111 Reserved	RW	3'b000
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid0	Valid programming for CPAG, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[2:0]	num_cxg_pag0	<p>Specifies the number of CXRAs in CPAGO Constraint: May use pag_tgid0 through pag_tgid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p>3'b000 1 port used</p> <p>3'b001 2 ports used</p> <p>3'b010 4 ports used</p> <p>3'b011 8 ports used</p> <p>3'b100 16 ports used</p> <p>3'b101 32 ports used</p> <p>3'b110 3 ports (MOD-3 hash)</p> <p>3'b111 Reserved</p>	RW	3'b000

5.2.4.102 cmn_hns_cml_port_aggr_ctrl_reg1-6

There are 6 iterations of this register. The index ranges from 1 to 6. Configures the CCIX port aggregation port groups.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(1-6) : 16'h6200 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.sam_control` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.sam_control` bit and `cmn_hns_scr.sam_control` bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-246: cmn_hns_cml_port_aggr_ctrl_reg1-6

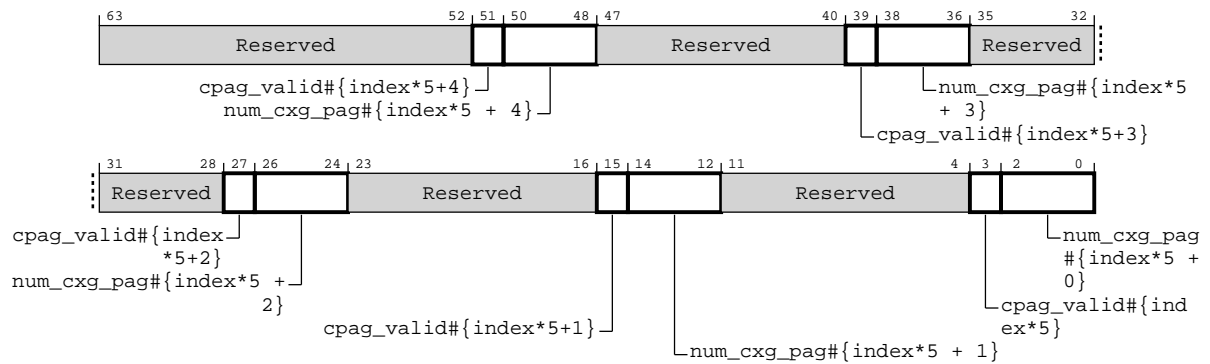


Table 5-250: cmn_hns_cml_port_aggr_ctrl_reg1-6 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	<code>cpag_valid#{index*5+4}</code>	Valid programming for CPAG <code>#{index*5 + 4}</code> , Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[50:48]	num_cxg_pag#{index*5 + 4}	<p>Specifies the number of CXRAs in CPAG4#{index*5 + 4} Constraint: May use pag_tgtid8 through pag_tgtid9 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p>3'b000 1 port used</p> <p>3'b001 2 ports used</p> <p>3'b010 4 ports used</p> <p>3'b011 8 ports used</p> <p>3'b100 16 ports used</p> <p>3'b101 32 ports used</p> <p>3'b110 3 ports (MOD-3 hash)</p> <p>3'b111 Reserved</p>	RW	3'b000
[47:40]	Reserved	Reserved	RO	-
[39]	cpag_valid#{index*5+3}	Valid programming for CPAG #{index*5 + 3}, Enabled by default (backward compatible)	RW	1'b1
[38:36]	num_cxg_pag#{index*5 + 3}	<p>Specifies the number of CXRAs in CPAG3#{index*5 + 3} Constraint: May use pag_tgtid6 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg1 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p>3'b000 1 port used</p> <p>3'b001 2 ports used</p> <p>3'b010 4 ports used</p> <p>3'b011 8 ports used</p> <p>3'b100 16 ports used</p> <p>3'b101 32 ports used</p> <p>3'b110 3 ports (MOD-3 hash)</p> <p>3'b111 Reserved</p>	RW	3'b000
[35:28]	Reserved	Reserved	RO	-
[27]	cpag_valid#{index*5+2}	Valid programming for CPAG #{index*5 + 2}, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[26:24]	num_cxg_pag#{index*5 + 2}	<p>Specifies the number of CXRAs in CPAG2#{index*5 + 2} Constraint: May use pag_tgid4 through pag_tgid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p>3'b000 1 port used</p> <p>3'b001 2 ports used</p> <p>3'b010 4 ports used</p> <p>3'b011 8 ports used</p> <p>3'b100 16 ports used</p> <p>3'b101 32 ports used</p> <p>3'b110 3 ports (MOD-3 hash)</p> <p>3'b111 Reserved</p>	RW	3'b000
[23:16]	Reserved	Reserved	RO	-
[15]	cpag_valid#{index*5+1}	Valid programming for CPAG #{index*5 + 1}, Enabled by default (backward compatible)	RW	1'b1
[14:12]	num_cxg_pag#{index*5 + 1}	<p>Specifies the number of CXRAs in CPAG1#{index*5 + 1} Constraint: May use pag_tgid2 through pag_tgid3 of cmn_hns_cml_port_aggr_grp_reg0 when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p>3'b000 1 port used</p> <p>3'b001 2 ports used</p> <p>3'b010 4 ports used</p> <p>3'b011 8 ports used</p> <p>3'b100 16 ports used</p> <p>3'b101 32 ports used</p> <p>3'b110 3 ports (MOD-3 hash)</p> <p>3'b111 Reserved</p>	RW	3'b000
[11:4]	Reserved	Reserved	RO	-
[3]	cpag_valid#{index*5}	Valid programming for CPAG #{index*5}, Enabled by default (backward compatible)	RW	1'b1

Bits	Name	Description	Type	Reset
[2:0]	num_cxg_pag#{index*5 + 0}	<p>Specifies the number of CXRAs in CPAG#{index*5 + 0} Constraint: May use pag_tgtid0 through pag_tgtid7 of cmn_hns_cml_port_aggr_grp_reg[0,1] when POR_RNSAM_FLEX_TGTID_EN_PARAM = 0</p> <p>3'b000 1 port used</p> <p>3'b001 2 ports used</p> <p>3'b010 4 ports used</p> <p>3'b011 8 ports used</p> <p>3'b100 16 ports used</p> <p>3'b101 32 ports used</p> <p>3'b110 3 ports (MOD-3 hash)</p> <p>3'b111 Reserved</p>	RW	3'b000

5.2.4.103 cmn_hns_abf_lo_addr

Lower address range for Address Based Flush (ABF) [51:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF50

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ppu

Secure group override

cmn_hns_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ppu bit and cmn_hns_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-247: cmn_hns_abf_lo_addr

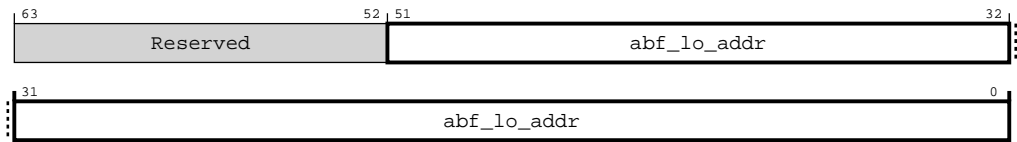


Table 5-251: cmn_hns_abf_lo_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:0]	abf_lo_addr	Lower address range for ABF	RW	52'b0

5.2.4.104 cmn_hns_abf_hi_addr

Upper address range for Address Based Flush (ABF) [51:0].

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF58

Type

RW

Reset value

See individual bit resets

Root group override
cmn_hns_rcr.ppu

Secure group override
cmn_hns_scr.ppu

Usage constraints
This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.ppu bit is set, Secure accesses to this register are permitted.
If both the cmn_hns_rcr.ppu bit and cmn_hns_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.
Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions
The following image shows the higher register bit assignments.

Figure 5-248: cmn_hns_abf_hi_addr

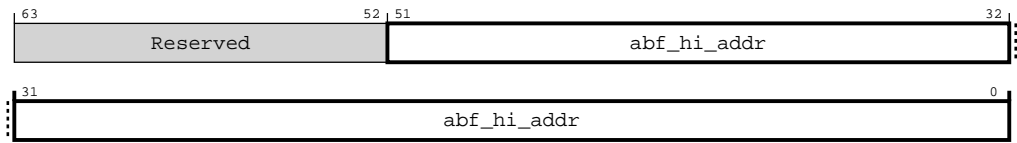


Table 5-252: cmn_hns_abf_hi_addr attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:0]	abf_hi_addr	Upper address range for ABF	RW	52'b0

5.2.4.105 cmn_hns_abf_pr

Functions as the Address Based Flush (ABF) policy register.

Configurations
This register is available in all configurations.

Attributes

Width
64

Address offset
16'hF60

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.ppu

Secure group override

cmn_hns_scr.ppu

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.ppu bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.ppu bit and cmn_hns_scr.ppu bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-249: cmn_hns_abf_pr



Table 5-253: cmn_hns_abf_pr attributes

Bits	Name	Description	Type	Reset
[63:10]	Reserved	Reserved	RO	-
[9:8]	abf_hbt_lbt_mode	ABF HBT/LBT Flush mode 2'b00: All addresses in ABF range (HBT and LBT) flushed 2'b01: All HBT addresses in ABF range flushed 2'b10: All LBT addresses in ABF range flushed 2'b11: Reserved	RW	2'b00
[7:3]	Reserved	Reserved	RO	-
[2:1]	abf_mode	ABF mode 2'b00: Clean Invalidate; WB dirty data and invalidate local copy 2'b01: Make Invalidate; invalidate without writing back dirty data 2'b10: Clean Shared; WB dirty data and can keep clean copy 2'b11: Reserved	RW	2'b00
[0]	abf_en	Start Address Based Flushing based on high and low address ranges	RW	1'b0

5.2.4.106 cmn_hns_abf_sr

Functions as the Address Based Flush (ABF) status register.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hF68

Type

RO

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-250: cmn_hns_abf_sr

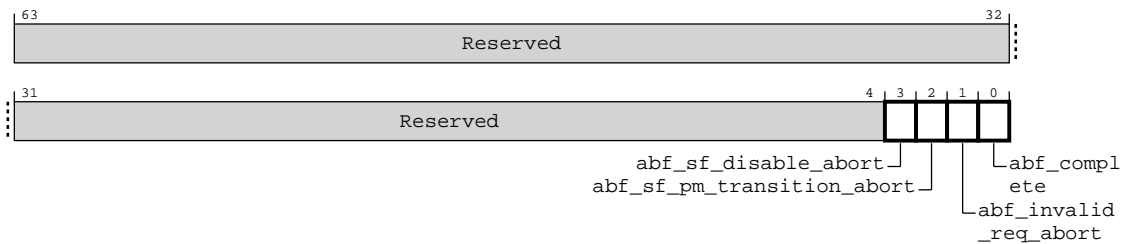


Table 5-254: cmn_hns_abf_sr attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3]	abf_sf_disable_abort	ABF aborted due to SF not being enabled, either by configuration or double-bit ECC error	RO	1'b0
[2]	abf_sf_pm_transition_abort	ABF aborted due to PM transition while ABF in progress, or both PM and ABF requested at the same time	RO	1'b0
[1]	abf_invalid_req_abort	ABF request made while PM is not in FAM/HAM/SF_ONLY mode; request aborted in this case	RO	1'b0
[0]	abf_complete	ABF completed	RO	1'b0

5.2.4.107 cmn_hns_cbusy_write_limit_ctl

Cbusy threshold limits for POCQ write entries. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1000

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

This register can be modified only with prior written permission from Arm.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-251: cmn_hns_cbusy_write_limit_ctl

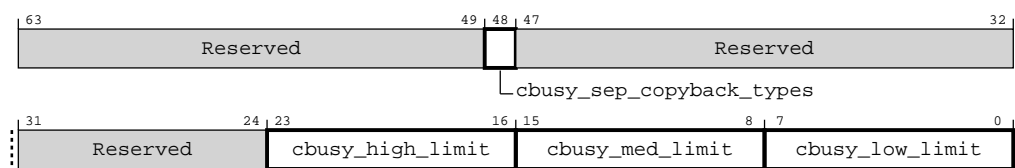


Table 5-255: cmn_hns_cbusy_write_limit_ctl attributes

Bits	Name	Description	Type	Reset
[63:49]	Reserved	Reserved	RO	-
[48]	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]	cbusy_high_limit	POCQ limit for Write CBusy High	RW	Configuration dependent
[15:8]	cbusy_med_limit	POCQ limit for Write CBusy Med	RW	Configuration dependent

Bits	Name	Description	Type	Reset
[7:0]	cbusy_low_limit	POCQ limit for Write CBusy Low	RW	Configuration dependent

5.2.4.108 cmn_hns_cbusy_resp_ctl

Controls the responses sent from HNS to RNF. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1008

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-252: cmn_hns_cbusy_resp_ctl

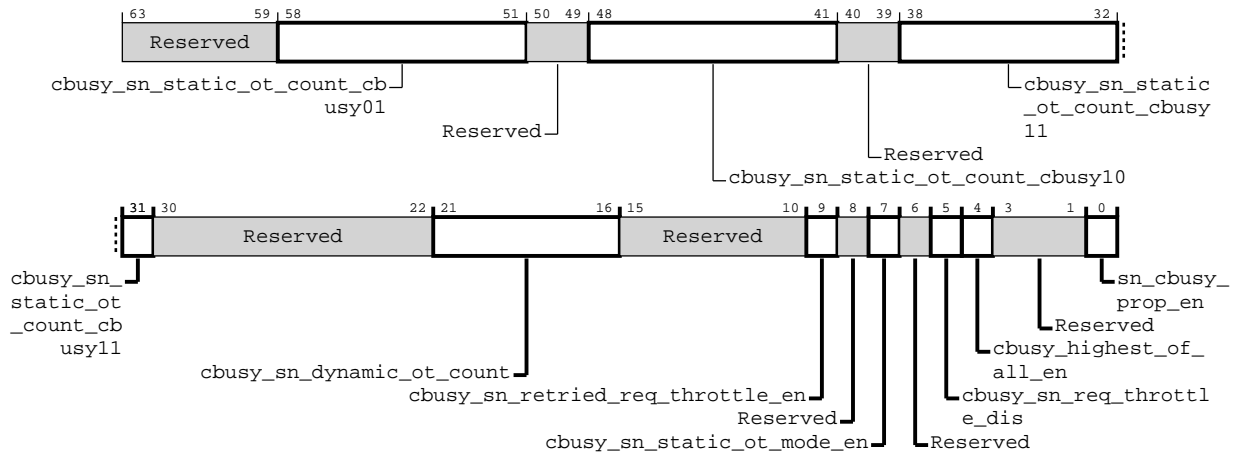


Table 5-256: cmn_hns_cbusy_resp_ctl attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:51]	cbusy_sn_static_ot_count_cbusy01	Specifies the maximum number of transactions to SN-F when SN Cbusy=01 in static throttling mode. CONSTRAINT: Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1	RW	Configuration dependent
[50:49]	Reserved	Reserved	RO	-
[48:41]	cbusy_sn_static_ot_count_cbusy10	Specifies the maximum number of transactions to SN-F when SN Cbusy=10 in static throttling mode. CONSTRAINT: Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1 and less than cbusy_sn_static_ot_count_cbusy01	RW	Configuration dependent
[40:39]	Reserved	Reserved	RO	-
[38:31]	cbusy_sn_static_ot_count_cbusy11	Specifies the maximum number of transactions to SN-F when SN Cbusy=11 in static throttling mode. CONSTRAINT: Value must be less than HNS_NUM_ENTRIES_POCQ_PARAM-1 and less than cbusy_sn_static_ot_count_cbusy10	RW	Configuration dependent
[30:22]	Reserved	Reserved	RO	-
[21:16]	cbusy_sn_dynamic_ot_count	Specifies the granularity at which HN-F will dynamically throttle transactions to SN-F. CONSTRAINT: 1,2,4,8 are the the allowed values	RW	6'b000100
[15:10]	Reserved	Reserved	RO	-
[9]	cbusy_sn_retried_req_throttle_en	Enables throttling retried requests with static grants (from SN) along with dynamic credit requests	RW	1'b0
[8]	Reserved	Reserved	RO	-
[7]	cbusy_sn_static_ot_mode_en	Controls cbusy between HN-F and SN-F 1'b0: HN-F will dynamically throttle outstanding requests to SN-F 1'b1: HN-F will use fixed transactions count at each CBusy level at 1/4th POCQ granularity CONSTRAINT: For SN request throttling, CBusy aggregation is always based on SN_CBusy[1:0] and cbusy_alt_mode_en is inapplicable	RW	1'b0
[6]	Reserved	Reserved	RO	-
[5]	cbusy_sn_req_throttle_dis	Disables Cbusy based request throttling from HNS to SNF when set to 1'b1	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	cbusy_highest_of_all_en	Controls cbusy between HN-F and SN-F 1'b0: Will send the HN-F or SN-F as configured 1'b1: Will select highest CBusy value between the SN-F and HN-F	RW	1'b0
[3:1]	Reserved	Reserved	RO	-
[0]	sn_cbusy_prop_en	Controls HN-F and SN-F cbusy on responses to RN-F 1'b0: HN-F's POCQ Cbusy is sent 1'b1: SN-F's Cbusy is sent	RW	1'b0

5.2.4.109 cmn_hns_cbusy_sn_ctl

Controls the SN-F cbusy thresholds. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1010

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-253: cmn_hns_cbusy_sn_ctl

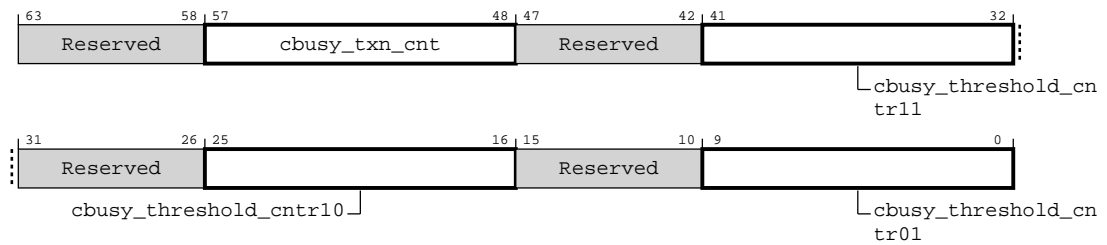


Table 5-257: cmn_hns_cbusy_sn_ctl attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57:48]	cbusy_txn_cnt	Number of transactions over which the counters are tracked	RW	10'b0100000000
[47:42]	Reserved	Reserved	RO	-
[41:32]	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]	cbusy_threshold_cntr10	CBusy threshold at which SN-F is considered busy for Counter_10	RW	10'b0000100000
[15:10]	Reserved	Reserved	RO	-
[9:0]	cbusy_threshold_cntr01	CBusy threshold at which SN-F is considered busy for Counter_01	RW	10'b0001000000

5.2.4.110 cmn_hns_lbt_cbusy_ctl

Controls the CBusy response for LCN Bound Transactions. CONSTRAINT: The hns_adv_cbusy_mode_en must be 1'b1 to use this feature.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h1018

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-254: cmn_hns_lbt_cbusy_ctl

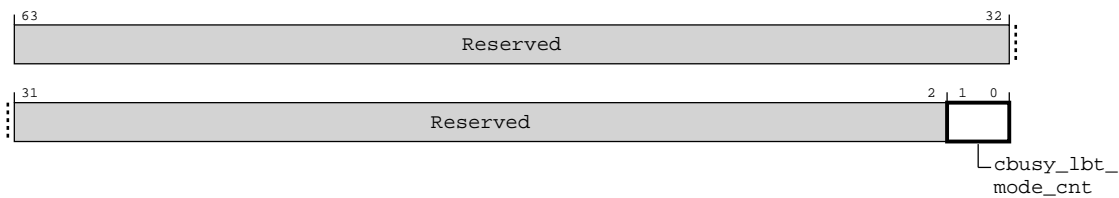


Table 5-258: cmn_hns_lbt_cbusy_ctl attributes

Bits	Name	Description	Type	Reset
[63:2]	Reserved	Reserved	RO	-
[1:0]	cbusy_lbt_mode_cnt	Controls the propagation of Cbusy field for LCN bound transactions. 2'b00: Send HNS POCQ Cbusy on all responses based on the limits programmed in cmn_hns_cbusy_limit_ctl 2'b01: Pass through HNF CBusy on late completion responses (CompData, Comp) 2'b10: Greater of POCQ Cbusy or HNF Cbusy. Applicable to responses where remote Cbusy can be sent	RW	2'b00

5.2.4.111 cmn_hns_pocq_alloc_class_dedicated

Controls Dedicated entries in POCQ for each class.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1020

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-255: cmn_hns_pocq_alloc_class_dedicated

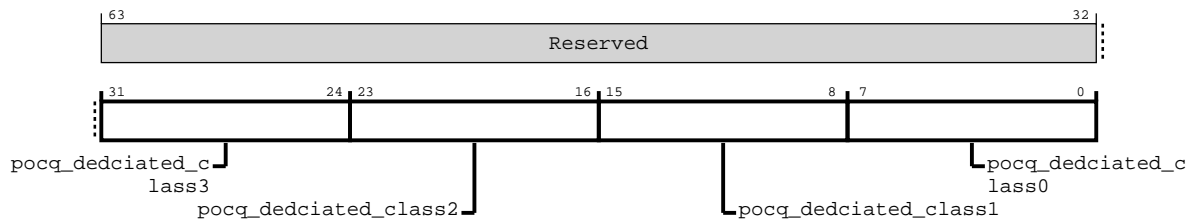


Table 5-259: cmn_hns_pocq_alloc_class_dedicated attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	pocq_dedciated_class3	Dedicated number of entries for Class 3 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class3 < hns_pocq_max_allowed_class3	RW	8'b00000000
[23:16]	pocq_dedciated_class2	Dedicated number of entries for Class 2 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class2 < hns_pocq_max_allowed_class2	RW	8'b00000000
[15:8]	pocq_dedciated_class1	Dedicated number of entries for Class 1 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class1 < hns_pocq_max_allowed_class1	RW	8'b00000000
[7:0]	pocq_dedciated_class0	Dedicated number of entries for Class 0 in POCQ CONSTRAINT: Sum of dedicated entries for classes & SEQ can not exceed HNS_NUM_ENTRIES_POCQ_PARAM. CONSTRAINT: hns_pocq_dedciated_class0 < hns_pocq_max_allowed_class0	RW	8'b00000000

5.2.4.112 cmn_hns_pocq_alloc_class_max_allowed

Controls Maximum allowed entries in POCQ for each class.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1028

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-256: cmn_hns_pocq_alloc_class_max_allowed

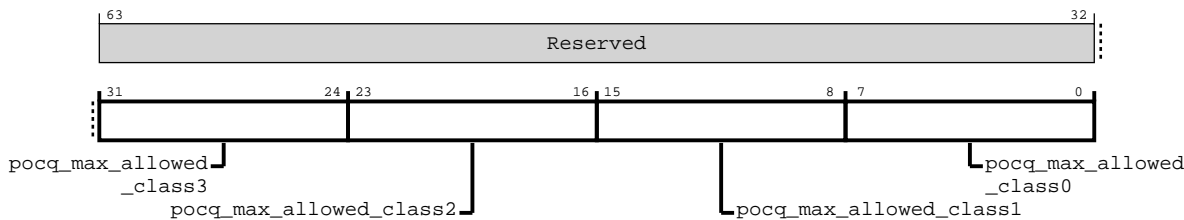


Table 5-260: cmn_hns_pocq_alloc_class_max_allowed attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[31:24]	pocq_max_allowed_class3	Maximum number of entries for Class 3 in POCQ CONSTRAINT: hns_pocq_dedciated_class3 < hns_pocq_max_allowed_class3	RW	Configuration dependent
[23:16]	pocq_max_allowed_class2	Maximum number of entries for Class 2 in POCQ CONSTRAINT: hns_pocq_dedciated_class2 < hns_pocq_max_allowed_class2	RW	Configuration dependent
[15:8]	pocq_max_allowed_class1	Maximum number of entries for Class 1 in POCQ CONSTRAINT: hns_pocq_dedciated_class1 < hns_pocq_max_allowed_class1	RW	Configuration dependent
[7:0]	pocq_max_allowed_class0	Maximum number of entries for Class 0 in POCQ CONSTRAINT: hns_pocq_dedciated_class0 < hns_pocq_max_allowed_class0	RW	Configuration dependent

5.2.4.113 cmn_hns_pocq_alloc_class_contended_min

Controls Contended minimum entries in POCQ for each class.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1030

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-257: cmn_hns_pocq_alloc_class_contended_min

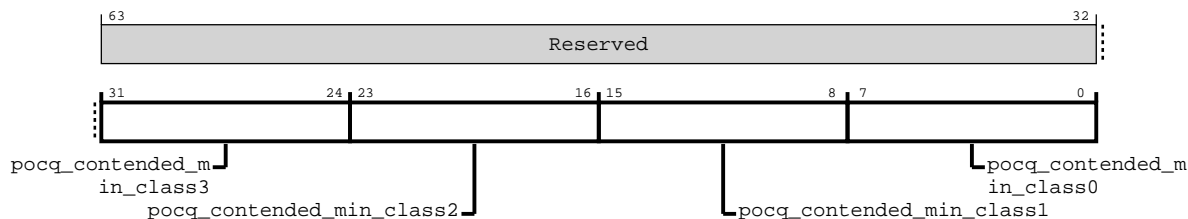


Table 5-261: cmn_hns_pocq_alloc_class_contended_min attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:24]	pocq_contended_min_class3	Contended min entries for Class 3 in POCQ	RW	Configuration dependent
[23:16]	pocq_contended_min_class2	Contended min entries for Class 2 in POCQ	RW	Configuration dependent
[15:8]	pocq_contended_min_class1	Contended min entries for Class 1 in POCQ	RW	Configuration dependent
[7:0]	pocq_contended_min_class0	Contended min entries for Class 0 in POCQ	RW	Configuration dependent

5.2.4.114 cmn_hns_pocq_alloc_misc_max_allowed

Controls Maximum allowed entries in POCQ for SNP, SEQ, and other misc req.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1038

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.qos` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.qos` bit and `cmn_hns_scr.qos` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-258: `cmn_hns_pocq_alloc_misc_max_allowed`

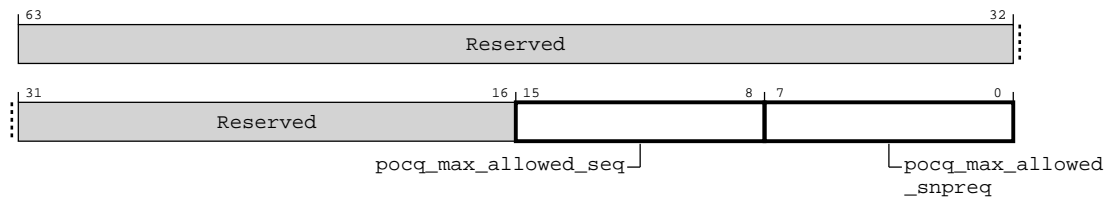


Table 5-262: `cmn_hns_pocq_alloc_misc_max_allowed` attributes

Bits	Name	Description	Type	Reset
[63:16]	Reserved	Reserved	RO	-
[15:8]	pocq_max_allowed_seq	Maximum number of entries for SEQ in POCQ. Constraint: Only values of 1 or 2 supported.	RW	8'h02
[7:0]	pocq_max_allowed_snpreq	Maximum number of entries for RXSNP requests in POCQ	RW	8'h04

5.2.4.115 `cmn_hns_class_ctl`

Class misc controls.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1040

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.qos`

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-259: cmn_hns_class_ctl

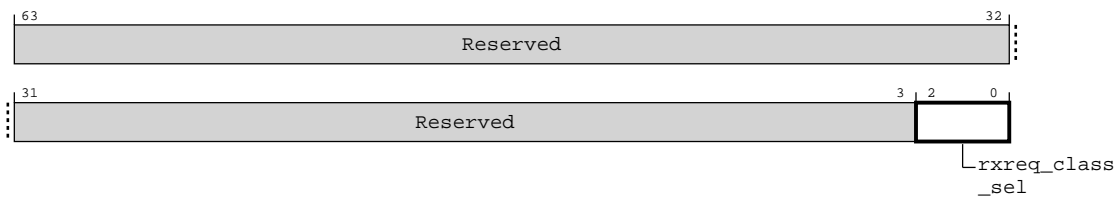


Table 5-263: cmn_hns_class_ctl attributes

Bits	Name	Description	Type	Reset
[63:3]	Reserved	Reserved	RO	-
[2:0]	rxreq_class_sel	RxReq Class select: 3'b000: QoS based class selection 3'b001: Request Opcode based class selection Note: If un-supported value is programmed, default selection of QoS based is chosen.	RW	3'b000

5.2.4.116 cmn_hns_pocq_qos_class_ctl

QoS bases class identification controls.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1048

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-260: cmn_hns_pocq_qos_class_ctl

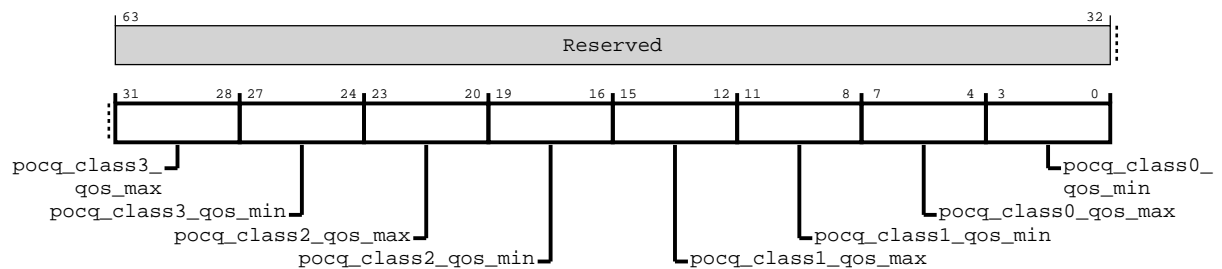


Table 5-264: cmn_hns_pocq_qos_class_ctl attributes

Bits	Name	Description	Type	Reset
[63:32]	Reserved	Reserved	RO	-
[31:28]	pocq_class3_qos_max	QoS maximum value for Class 3	RW	4'b0111
[27:24]	pocq_class3_qos_min	QoS minimum value for Class 3	RW	4'b0000
[23:20]	pocq_class2_qos_max	QoS maximum value for Class 2	RW	4'b1011
[19:16]	pocq_class2_qos_min	QoS minimum value for Class 2	RW	4'b1000
[15:12]	pocq_class1_qos_max	QoS maximum value for Class 1	RW	4'b1110
[11:8]	pocq_class1_qos_min	QoS minimum value for Class 1	RW	4'b1100
[7:4]	pocq_class0_qos_max	QoS maximum value for Class 0	RW	4'b1111
[3:0]	pocq_class0_qos_min	QoS minimum value for Class 0	RW	4'b1111

5.2.4.117 cmn_hns_class_pocq_arb_weight_ctl

Per Class weight controls for scheduling requests from POCQ.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1050

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-261: cmn_hns_class_pocq_arb_weight_ctl

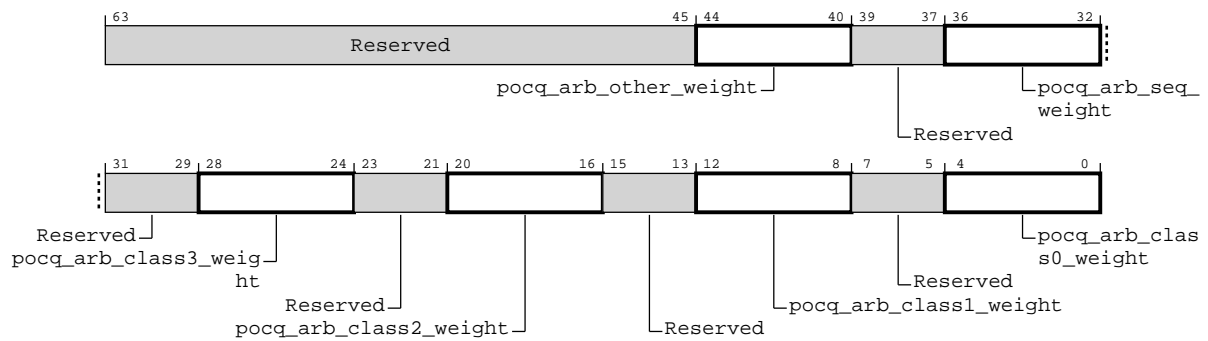


Table 5-265: cmn_hns_class_pocq_arb_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:40]	pocq_arb_other_weight	Other req weight for scheduling requests from POCQ	RW	5'b00000
[39:37]	Reserved	Reserved	RO	-
[36:32]	pocq_arb_seq_weight	SEQ weight for scheduling requests from POCQ	RW	5'b00000
[31:29]	Reserved	Reserved	RO	-
[28:24]	pocq_arb_class3_weight	Class3 weight for scheduling requests from POCQ	RW	5'b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	pocq_arb_class2_weight	Class2 weight for scheduling requests from POCQ	RW	5'b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	pocq_arb_class1_weight	Class1 weight for scheduling requests from POCQ	RW	5'b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	pocq_arb_class0_weight	Class0 weight for scheduling requests from POCQ	RW	5'b00000

5.2.4.118 cmn_hns_class_retry_weight_ctl

Per Class weight controls for Retry Credit grant.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1058

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.qos` bit and `cmn_hns_scr.qos` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-262: `cmn_hns_class_retry_weight_ctl`

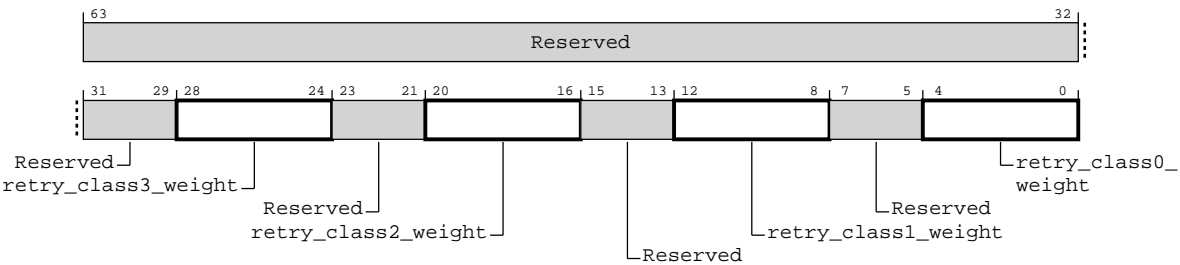


Table 5-266: `cmn_hns_class_retry_weight_ctl` attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:24]	<code>retry_class3_weight</code>	Overall Class3 weight for credit grant arbitration	RW	5'b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	<code>retry_class2_weight</code>	Overall Class2 weight for credit grant arbitration	RW	5'b00000
[15:13]	Reserved	Reserved	RO	-
[12:8]	<code>retry_class1_weight</code>	Overall Class1 weight for credit grant arbitration	RW	5'b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	<code>retry_class0_weight</code>	Overall Class0 weight for credit grant arbitration	RW	5'b00000

5.2.4.119 `cmn_hns_pocq_misc_retry_weight_ctl`

Weight controls for Snoop, SEQ, Flush and other misc POCQ requests.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h1060

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.qos

Secure group override

cmn_hns_scr.qos

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.qos bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.qos bit and cmn_hns_scr.qos bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-263: cmn_hns_pocq_misc_retry_weight_ctl

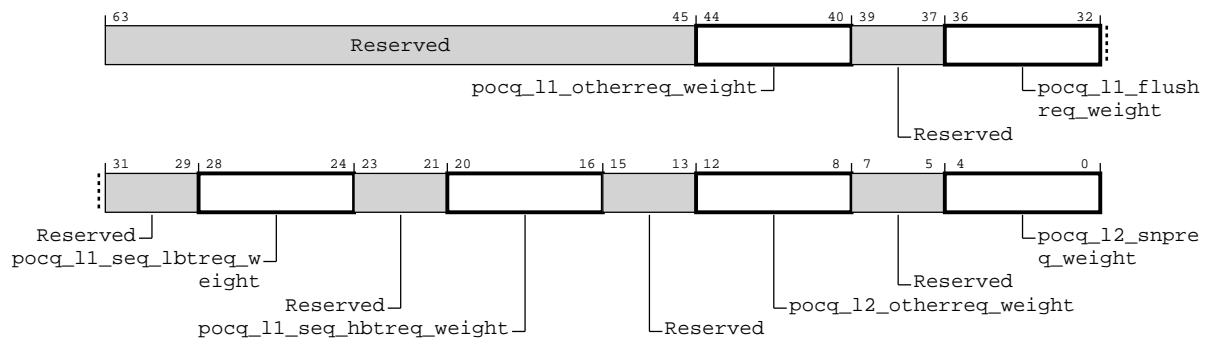


Table 5-267: cmn_hns_pocq_misc_retry_weight_ctl attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:40]	pocq_l1_otherreq_weight	Weight for other requests (Ex: Debug Read) for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000
[39:37]	Reserved	Reserved	RO	-
[36:32]	pocq_l1_flushreq_weight	Weight for SLF/SF Flush requests for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000
[31:29]	Reserved	Reserved	RO	-
[28:24]	pocq_l1_seq_lbtreq_weight	Weight for SEQ-LBT requests for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000
[23:21]	Reserved	Reserved	RO	-
[20:16]	pocq_l1_seq_hbtreq_weight	Weight for SEQ-HBT requests for POCQ allocation arbitration for Level 1. Note: This is first level arb weight control. Second level after this arb is for snpreq.	RW	5'b00000

Bits	Name	Description	Type	Reset
[15:13]	Reserved	Reserved	RO	-
[12:8]	pocq_l2_otherreq_weight	Weight for other requests (Ex: Level 1 arb req) for POCQ allocation arbitration for Level 2. Note: This is second level arb weight control. First level is seq, flush, dbgrd, etc.	RW	5'b00000
[7:5]	Reserved	Reserved	RO	-
[4:0]	pocq_l2_snpreq_weight	Weight for external snoop requests for POCQ allocation arbitration for Level 2. Note: This is second level arb weight control. First level is seq, flush, dbgrd, etc	RW	5'b00000

5.2.4.120 cmn_hns_partner_scratch_reg0

Partner scratch register 0

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hFEO

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.partner_scratch_override

Secure group override

cmn_hns_scr.partner_scratch_override

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.partner_scratch_override bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.partner_scratch_override bit and cmn_hns_scr.partner_scratch_override bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-264: cmn_hns_partner_scratch_reg0

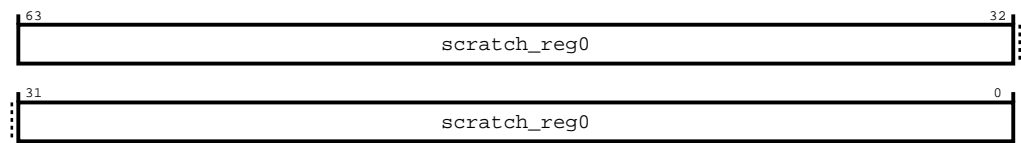


Table 5-268: cmn_hns_partner_scratch_reg0 attributes

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg0	64 bit scratch register 0 wirh read/write access	RW	64'h00000000

5.2.4.121 cmn_hns_partner_scratch_reg1

Partner scratch register 1

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hFE8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.partner_scratch_override

Secure group override

cmn_hns_scr.partner_scratch_override

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.partner_scratch_override bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.partner_scratch_override bit and cmn_hns_scr.partner_scratch_override bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-265: cmn_hns_partner_scratch_reg1

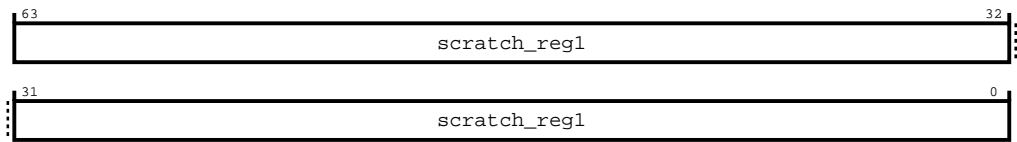


Table 5-269: cmn_hns_partner_scratch_reg1 attributes

Bits	Name	Description	Type	Reset
[63:0]	scratch_reg1	64 bit scratch register 1 with read/write access	RW	64'h00000000

5.2.4.122 cmn_hns_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

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:
If the cmn_hns_rcr.slcsf_dbgrd bit is set, Secure accesses to this register are permitted.
If both the cmn_hns_rcr.slcsf_dbgrd bit and cmn_hns_scr.slcsf_dbgrd bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-266: cmn_hns_cfg_slcsf_dbgrd

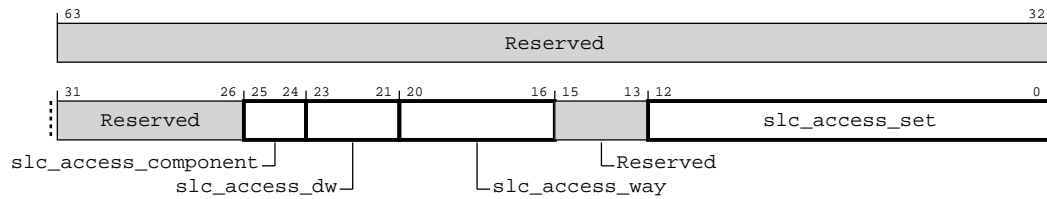


Table 5-270: cmn_hns_cfg_slcsf_dbgrd attributes

Bits	Name	Description	Type	Reset
[63:26]	Reserved	Reserved	RO	-
[25:24]	slc_access_component	Specifies SLC/SF array debug read 2'b01: SLC data read 2'b10: SLC tag read 2'b11: SF tag read	WO	2'b00
[23:21]	slc_access_dw	64-bit chunk address for SLC data debug read access	WO	3'h0
[20:16]	slc_access_way	Way address for SLC/SF debug read access	WO	5'h00
[15:13]	Reserved	Reserved	RO	-
[12:0]	slc_access_set	Set address for SLC/SF debug read access	WO	13'h0

5.2.4.123 cmn_hns_slc_cache_access_slc_tag

Contains SLC tag debug read data bits [63:0]

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hB88

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-267: cmn_hns_slc_cache_access_slc_tag

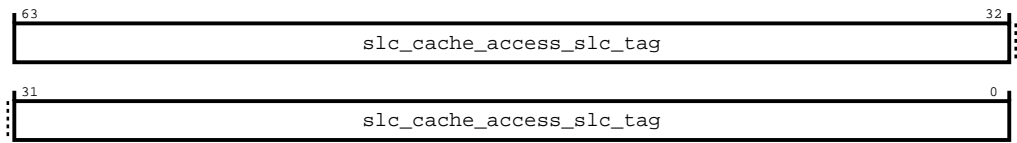


Table 5-271: cmn_hns_slc_cache_access_slc_tag attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_tag	SLC tag debug read data	RO	64'h0

5.2.4.124 cmn_hns_slc_cache_access_slc_tag1

Contains SLC tag debug read data bits [127:64] when present

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hB90

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-268: cmn_hns_slc_cache_access_slc_tag1

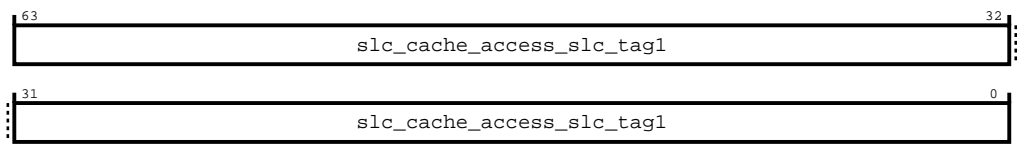


Table 5-272: cmn_hns_slc_cache_access_slc_tag1 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_tag1	SLC tag debug read data	RO	64'h0

5.2.4.125 cmn_hns_slc_cache_access_slc_data

Contains SLC data RAM debug read data.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hB98

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-269: cmn_hns_slc_cache_access_slc_data

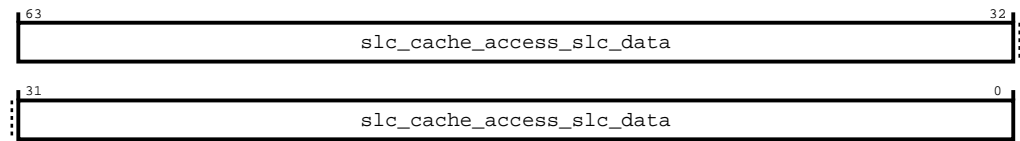


Table 5-273: cmn_hns_slc_cache_access_slc_data attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_slc_data	SLC data RAM debug read data	RO	64'h0

5.2.4.126 cmn_hns_slc_cache_access_slc_mte_tag

Contains MTE Tag data for the corresponding SLC data RAM debug read.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hBC0

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-270: cmn_hns_slc_cache_access_slc_mte_tag

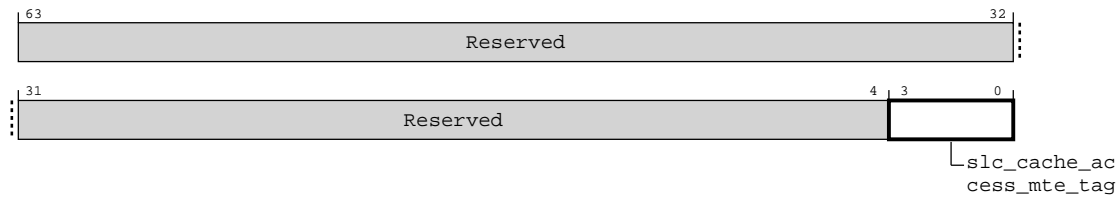


Table 5-274: cmn_hns_slc_cache_access_slc_mte_tag attributes

Bits	Name	Description	Type	Reset
[63:4]	Reserved	Reserved	RO	-
[3:0]	slc_cache_access_mte_tag	SLC MTE TAG corresponding to data RAM debug read data (128bit chunk of data)	RO	4'h0

5.2.4.127 cmn_hns_slc_cache_access_sf_tag

Contains SF tag debug read data. Bits[63:0]

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hBA0

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-271: cmn_hns_slc_cache_access_sf_tag

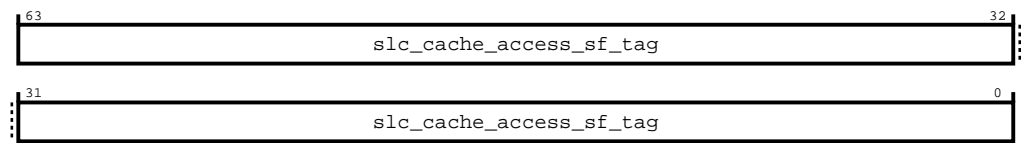


Table 5-275: cmn_hns_slc_cache_access_sf_tag attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag	SF tag debug read data	RO	64'h0

5.2.4.128 cmn_hns_slc_cache_access_sf_tag1

Contains SF tag debug read data bits [127:64], when present in SF Tag

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hBA8

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-272: cmn_hns_slc_cache_access_sf_tag1

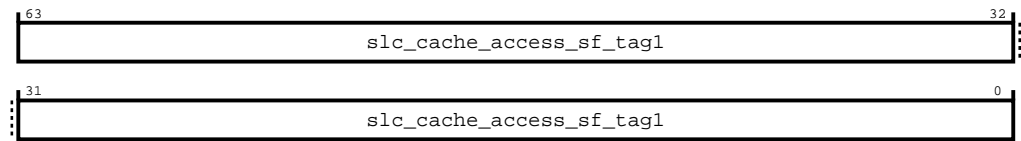


Table 5-276: cmn_hns_slc_cache_access_sf_tag1 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag1	SF tag debug read data	RO	64'h0

5.2.4.129 cmn_hns_slc_cache_access_sf_tag2

Contains SF tag debug read data bits [128:191], when present in SF Tag

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hBB0

Type

RO

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.slcsf_dbgrd

Secure group override

cmn_hns_scr.slcsf_dbgrd

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-273: cmn_hns_slc_cache_access_sf_tag2

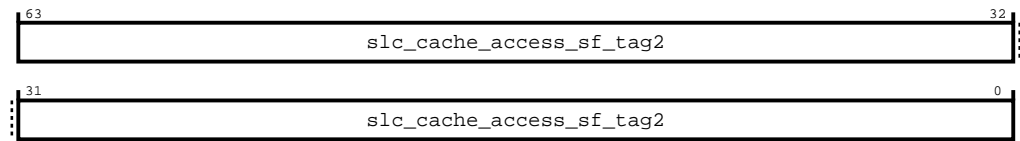


Table 5-277: cmn_hns_slc_cache_access_sf_tag2 attributes

Bits	Name	Description	Type	Reset
[63:0]	slc_cache_access_sf_tag2	SF tag debug read data	RO	64'h0

5.2.4.130 cmn_hns_pmu_event_sel

Specifies the PMU event to be counted.

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'hD900

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-274: cmn_hns_pmu_event_sel

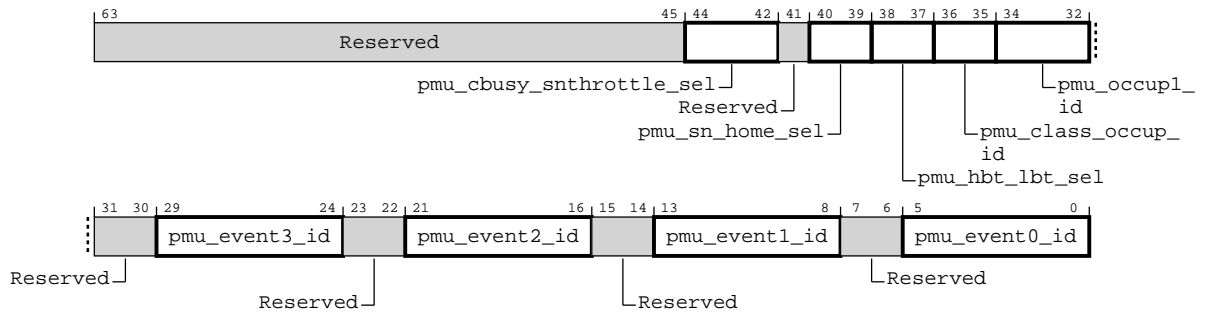


Table 5-278: cmn_hns_pmu_event_sel attributes

Bits	Name	Description	Type	Reset
[63:45]	Reserved	Reserved	RO	-
[44:42]	pmu_cbusy_snthrottle_sel	Filter for selecting specific SN throttle type 3'b000: All SN types throttled 3'b001: SN Group 0 Reads 3'b010: SN Group 0 Non-Reads 3'b011: SN Group 1 Reads 3'b100: SN Group 1 Non-Reads 3'b101: All SN Reads 3'b110: All SN Non-Reads	RW	3'h0
[41]	Reserved	Reserved	RO	-
[40:39]	pmu_sn_home_sel	HN-F PMU SN/Home select 2'b00: All requests selected 2'b01: SN bound requests selected 2'b10: Home bound requests selected	RW	2'h0
[38:37]	pmu_hbt_lbt_sel	HN-F PMU HBT/LBT select 2'b00: All requests selected 2'b01: HBT requests selected 2'b10: LBT requests selected	RW	2'h0
[36:35]	pmu_class_occup_id	HN-F PMU Class select 2'b00: Class 0 selected 2'b01: Class 1 selected 2'b10: Class 2 selected 2'b11: Class 3 selected	RW	2'h0
[34:32]	pmu_occup1_id	HN-F PMU occupancy 1 select 3'b000: All occupancy selected 3'b001: Read requests 3'b010: Write requests 3'b011: Atomic operation requests 3'b100: Stash requests 3'b101: RxSnp requests 3'b110: LBT requests 3'b111: HBT requests	RW	3'h0
[31:30]	Reserved	Reserved	RO	-
[29:24]	pmu_event3_id	HN-F PMU Event 3 select; see pmu_event0_id for encodings	RW	6'h00
[23:22]	Reserved	Reserved	RO	-
[21:16]	pmu_event2_id	HN-F PMU Event 2 select; see pmu_event0_id for encodings	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	pmu_event1_id	HN-F PMU Event 1 select; see pmu_event0_id for encodings	RW	6'h00
[7:6]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>HN-F PMU Event 0 select 6'h00: No event 6'h01: PMU_HN_CACHE_MISS_EVENT; counts total cache misses in first lookup result (high priority). Filtering is programmed in pmu_hbt_lbt_sel 6'h02: PMU_HN_SLCSF_CACHE_ACCESS_EVENT; counts number of cache accesses in first access (high priority). Filtering is programmed in pmu_hbt_lbt_sel 6'h03: PMU_HN_CACHE_FILL_EVENT; counts total allocations in HN SLC (all cache line allocations to SLC). Filtering is programmed in pmu_hbt_lbt_sel 6'h04: PMU_HN_POCQ_RETRY_EVENT; counts number of retried requests. Filtering is programmed in pmu_hbt_lbt_sel 6'h05: PMU_HN_POCQ_REQS_RECVD_EVENT; counts number of requests received by HN. Filtering is programmed in pmu_hbt_lbt_sel 6'h06: PMU_HN_SF_HIT_EVENT; counts number of SF hits. Filtering is programmed in pmu_hbt_lbt_sel 6'h07: PMU_HN_SF_EVICTIONS_EVENT; counts number of SF eviction cache invalidations initiated. Filtering is programmed in pmu_hbt_lbt_sel 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. Filtering is programmed in pmu_hbt_lbt_sel 6'h0B: PMU_HN_SLC_FILL_INVALID_WAY_EVENT; counts number of SLC fills to an invalid way. Filtering is programmed in pmu_hbt_lbt_sel 6'h0C: PMU_HN_MC_RETRIES_LOCAL_EVENT; counts number of local retried transactions by the MC 6'h0D: PMU_HN_MC_REQS_LOCAL_EVENT; counts number of local requests sent to MC. Filtering is programmed in pmu_sn_home_sel 6'h0E: PMU_HN_QOS_HH_RETRY_EVENT; counts number of times a HighHigh priority request is protocol retried at the HN-F 6'h0F: PMU_HN_POCQ_OCCUPANCY_EVENT; counts the POCQ occupancy in HN-F; occupancy filtering is programmed in pmu_occup1_id 6'h10: PMU_HN_POCQ_ADDRHAZ_EVENT; counts number of POCQ address hazards upon allocation 6'h11: PMU_HN_POCQ_ATOMICS_ADDRHAZ_EVENT; counts number of POCQ address hazards upon allocation for atomic operations 6'h12: PMU_HN_LD_ST_SWP_ADQ_FULL_EVENT; counts number of times ADQ is full for Ld/St/SWP type atomic operations while POCQ has pending operations 6'h13: PMU_HN_CMP_ADQ_FULL_EVENT; counts number of times ADQ is full for CMP type atomic operations while POCQ has pending operations 6'h14: PMU_HN_TXDAT_STALL_EVENT; counts number of times HN-F has a pending TXDAT flit but no credits to upload 6'h15: PMU_HN_TXRSP_STALL_EVENT; counts number of times HN-F has a pending TXRSP flit but no credits to upload 6'h16: PMU_HN_SEQ_FULL_EVENT; counts number of times requests are replayed in SLC pipe due to SEQ being full 6'h17: PMU_HN_SEQ_HIT_EVENT; counts number of times a request in SLC hit a pending SF eviction in SEQ 6'h18: PMU_HN_SNP_SENT_EVENT; counts number of snoops sent including directed/multicast/SF back invalidation 6'h19: PMU_HN_SFBI_DIR_SNP_SENT_EVENT; counts number of times directed snoops were sent due to SF back invalidation 6'h1a: PMU_HN_SFBI_BRD_SNP_SENT_EVENT; counts number of times multicast snoops were sent due to SF back invalidation 6'h1b: Reserved 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</p>	RW	6'h00

Bits	Name	Description	Type	Reset
[5:0]	pmu_event0_id	<p>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 individual snoops within a cluster 6'h24: PMU_HN_SF_IMPRECISE_EVICT_EVENT; counts number of times an evict op was dropped due to SF clustering 6'h25: PMU_HN_SF_EVICT_SHARED_LINE_EVENT; counts number of times a shared line was evicted from SF 6'h26: PMU_HN_POCQ_CLASS_OCCUPANCY_EVENT; counts the POCQ occupancy for a given class in HN-F; Class occupancy filtering is programmed in pmu_class_occup_id 6'h27: PMU_HN_POCQ_CLASS_RETRY_EVENT; counts number of retried requests for a given class; Class filtering is programmed in pmu_class_occup_id 6'h28: PMU_HN_CLASS_MC_REQS_LOCAL_EVENT; counts number of local requests sent to MC for a given class; Class filtering is programmed in pmu_class_occup_id 6'h29: PMU_HN_CLASS_PCRDGNT_BELOW_CONDMIN_EVENT; counts number of protocol credit grants for a given class when it's above dedicated and below conditional min; Class filtering is programmed in pmu_class_occup_id 6'h2A: PMU_HN_NUM_SN_CBUSY_THROTTLE_EVENT; counts number of times request to SN was throttled due to cbusy; Event filtering is programmed in pmu_cbusy_snthrottle_sel 6'h2B: PMU_HN_NUM_SN_CBUSY_THROTTLE_MIN_EVENT; counts number of times request to SN was throttled to the minimum allowed value of 4, due to cbusy; Event filtering is programmed in pmu_cbusy_snthrottle_sel 6'h2C: PMU_HN_SF_PRECISE_TO_IMPRECISE_EVENT; counts when number sharers exceeds how many RN's could be precisely tracked in SF 6'h2D: PMU_HN_SNP_INTV_CLN_EVENT; counts the number of times clean data intervened for a snoop request 6'h2E: PMU_HN_NC_EXCL_EVENT; counts the number of times non-cacheable exclusive request arrived at HNF 6'h2F: PMU_HN_EXCL_MON_OVFL_EVENT; counts the number of times exclusive monitor overflowed 6'h30: PMU_HN_SNP_REQ_RECVD_EVENT; counts number of incoming snoop requests 6'h31: PMU_HN_SNP_REQ_BYP_POCQ_EVENT; counts number of times incoming snoop request bypass Snoop Queue and allocates in POCQ 6'h32: PMU_HN_DIR_CCGHA_SNP_SENT_EVENT; counts number of directed snoops sent to CCG HA. (SFBI and non-SFBI included) 6'h33: PMU_HN_BRD_CCGHA_SNP_SENT_EVENT; counts number of broadcast snoops sent to CCG HA. (SFBI and non-SFBI included) 6'h34: PMU_HN_CCGHA_SNP_STALL_EVENT; counts number of times snoop is ready to be sent to CCG HA but stalled due to maximum snoops allowed. 6'h35: PMU_HN_LBT_REQ_OVER_HARDLIM_EVENT; counts number of times LBT write req can't allocate in SLC due to being over LBT cache capacity 6'h36: PMU_HN_HBT_REQ_OVER_HARDLIM_EVENT; counts number of times HBT write req can't allocate in SLC due to being over HBT cache capacity 6'h37: PMU_HN_SF_REUPDATE_EVENT; counts number of times SF needs to be re-updated 6'h38: PMU_HN_EXCL_SF_IMPRECISE_EVENT; counts number of times exclusive req hits SF in imprecise mode 6'h39: PMU_HN_SNP_POCQ_ADDRHAZ_EVENT; counts number of POCQ address hazards involving external snoop requests at allocation 6'h3A: PMU_HN_MC_RETRIES_REMOTE_EVENT; counts number of remote retried transactions by the MC 6'h3B: PMU_HN_MC_REQS_REMOTE_EVENT; counts number of remote requests sent to MC. Filtering is programmed in pmu_sn_home_sel 6'h3C: PMU_HN_CLASS_MC_REQS_REMOTE_EVENT; counts number of remote requests sent to MC for a given class; Class filtering is programmed in pmu_class_occup_id</p>	RW	6'h00

5.2.4.131 cmn_hns_pmu_mpam_sel

Specifies details of MPAM event to be counted

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'hD908

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-275: cmn_hns_pmu_mpam_sel

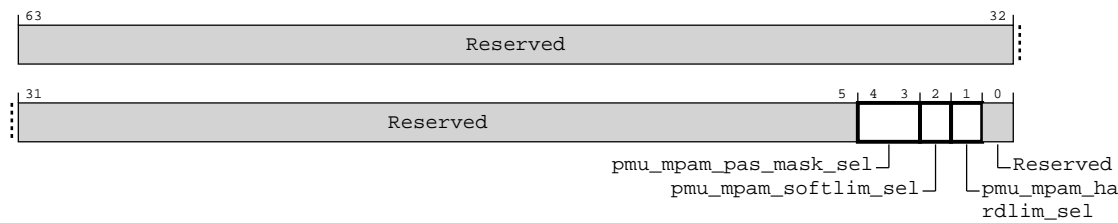


Table 5-279: cmn_hns_pmu_mpam_sel attributes

Bits	Name	Description	Type	Reset
[63:5]	Reserved	Reserved	RO	-
[4:3]	pmu_mpam_pas_mask_sel	PAS select for PARTID Mask 2'b00: PMU MPAM mask is for Secure MPAMID. 2'b01: PMU MPAM mask is for Non-Secure MPAMID. 2'b10: PMU MPAM mask is for Root MPAMID. 2'b11: PMU MPAM mask is for Realm MPAMID.	RW	2'b01
[2]	pmu_mpam_softlim_sel	When set, HN-F PMU MPAM Softlimit count is filtered for specific PARTIDs 1'b0: PMU Softlimit count is total for all PARDIDs. 1'b1: PMU Softlimit count is only for PARDIDs indicated in fliter register	RW	1'b0

Bits	Name	Description	Type	Reset
[1]	pmu_mpam_hardlim_sel	When set, HN-F PMU MPAM Hardlimit count is filtered for specific PARTIDs 1'b0: PMU Hardlimit count is total for all PARDIDs. 1'b1: PMU Hardlimit count is only for PARDIDs indicated in filter register	RW	1'b0
[0]	Reserved	Reserved	RO	-

5.2.4.132 cmn_hns_pmu_mpam_pardid_mask0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Functions as mask for PARTID[#{64*(index+1)-1};#{64*index}] filter for MPM PMU events

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

```
0x16'hD910 + #{8*index}
0xindex(0) : 16'h0 + (POR_CHI_MPAM_ENABLE_PARAM == 1)
0xindex(1) : 16'h0 + (HNS_MPAM_NS_PARTID_MAX_PARAM > 64) ||
(HNS_MPAM_S_PARTID_MAX_PARAM > 64)
0xindex(2-3) : 16'h0 + (HNS_MPAM_NS_PARTID_MAX_PARAM > 128) ||
(HNS_MPAM_S_PARTID_MAX_PARAM > 128)
0xindex(4-7) : 16'h0 + (HNS_MPAM_NS_PARTID_MAX_PARAM > 256) ||
(HNS_MPAM_S_PARTID_MAX_PARAM > 256)
```

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Non-Secure space and is accessible using Non-secure, Secure, Root, and Realm transactions.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-276: cmn_hns_pmu_mpam_pardid_mask0-7

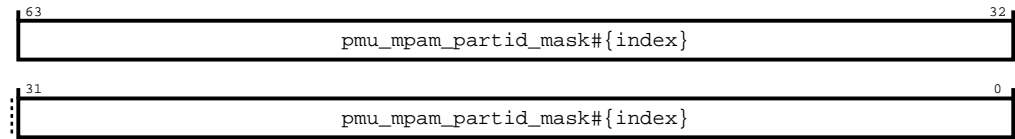


Table 5-280: cmn_hns_pmu_mpam_pardid_mask0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	pmu_mpam_partid_mask#{index}	MPAM PMU hardlimit and softlimit mask for PARTID [#{64*(index + 1)-1}:#{64*index}] 1'b0: PARTID specified is not counted in PMU count. 1'b1: PARTID specified is counted in PMU count. Note: This mask is used only when cmn_hns_pmu_mpam_sel is set for PARTID based counting.	RW	64'b0

5.2.4.133 cmn_hns_rn_cluster0-63_physid_reg0

There are 64 iterations of this register. The index ranges from 0 to 63. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h3C00 + #{index}*32

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.sam_control` bit and `cmn_hns_scr.sam_control` bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-277: `cmn_hns_rn_cluster0-63_physid_reg0`

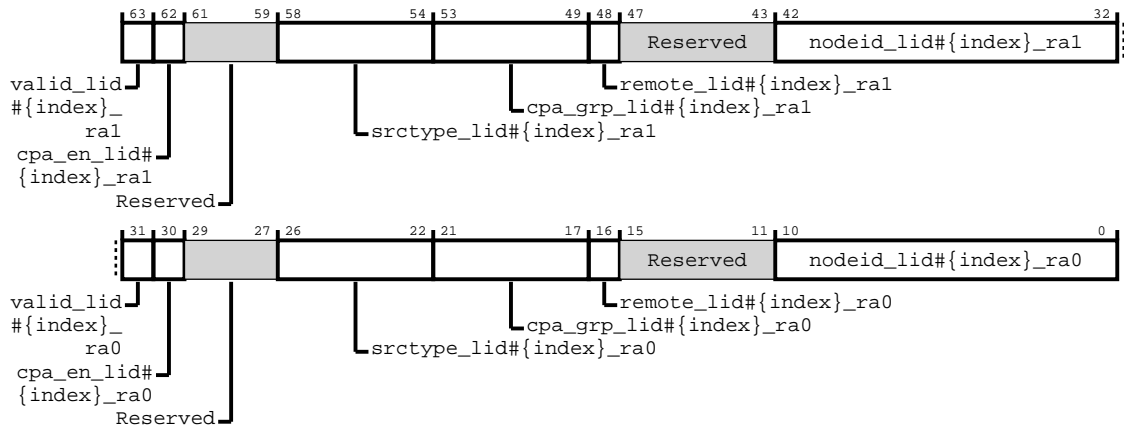


Table 5-281: `cmn_hns_rn_cluster0-63_physid_reg0` attributes

Bits	Name	Description	Type	Reset
[63]	<code>valid_lid#{index}_ra1</code>	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]	<code>cpa_en_lid#{index}_ra1</code>	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[61:59]	Reserved	Reserved	RO	-
[58:54]	<code>srctype_lid#{index}_ra1</code>	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[53:49]	<code>cpa_grp_lid#{index}_ra1</code>	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	<code>remote_lid#{index}_ra1</code>	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]	<code>nodeid_lid#{index}_ra1</code>	Specifies the node ID	RW	11'h0
[31]	<code>valid_lid#{index}_ra0</code>	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]	<code>cpa_en_lid#{index}_ra0</code>	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[29:27]	Reserved	Reserved	RO	-
[26:22]	<code>srctype_lid#{index}_ra0</code>	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[21:17]	<code>cpa_grp_lid#{index}_ra0</code>	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	<code>remote_lid#{index}_ra0</code>	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

5.2.4.134 cmn_hns_rn_cluster64-127_physid_reg0

There are 64 iterations of this register. The index ranges from 64 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h3C00 + #{index}*32

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-278: cmn_hns_rn_cluster64-127_physid_reg0

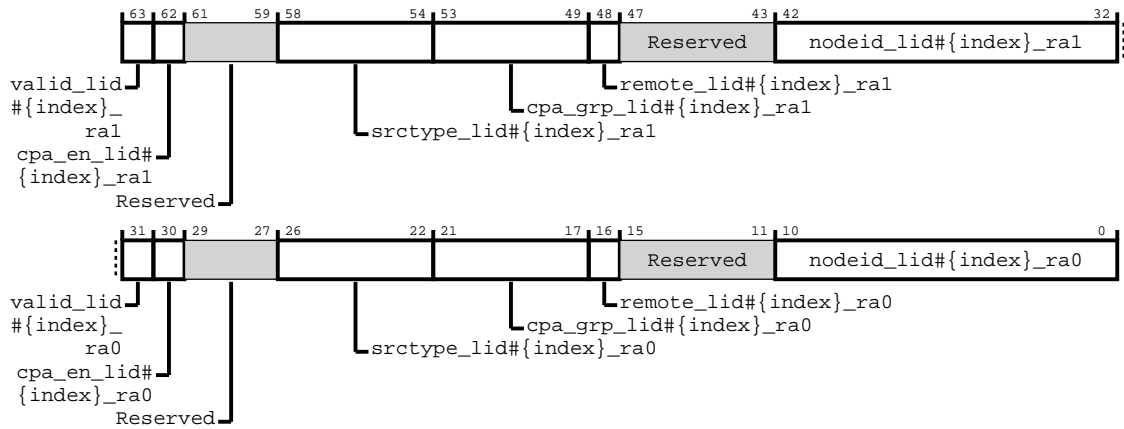


Table 5-282: cmn_hns_rn_cluster64-127_physid_reg0 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra1	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra1	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[61:59]	Reserved	Reserved	RO	-
[58:54]	srctype_lid#{index}_ra1	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[53:49]	cpa_grp_lid#{index}_ra1	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra1	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra1	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra0	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra0	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[29:27]	Reserved	Reserved	RO	-
[26:22]	srctype_lid#{index}_ra0	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[21:17]	cpa_grp_lid#{index}_ra0	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra0	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra0	Specifies the node ID	RW	11'h0

5.2.4.135 cmn_hns_rn_cluster0-127_physid_reg1

There are 128 iterations of this register. The index ranges from 0 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h3C08 + #{index}*32

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-279: cmn_hns_rn_cluster0-127_physid_reg1

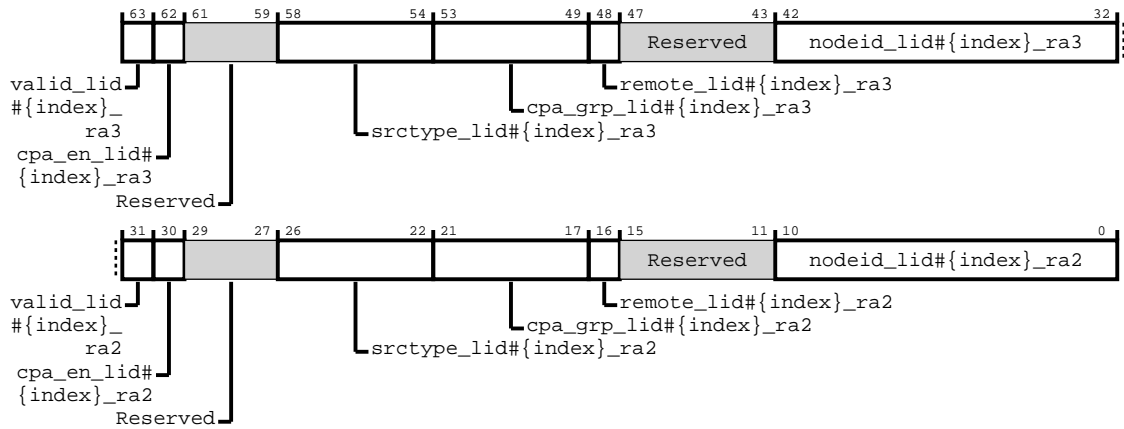


Table 5-283: cmn_hns_rn_cluster0-127_physid_reg1 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra3	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra3	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[61:59]	Reserved	Reserved	RO	-
[58:54]	srctype_lid#{index}_ra3	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[53:49]	cpa_grp_lid#{index}_ra3	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra3	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra3	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra2	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra2	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[29:27]	Reserved	Reserved	RO	-
[26:22]	srctype_lid#{index}_ra2	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[21:17]	cpa_grp_lid#{index}_ra2	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra2	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra2	Specifies the node ID	RW	11'h0

5.2.4.136 cmn_hns_rn_cluster0-127_physid_reg2

There are 128 iterations of this register. The index ranges from 0 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h3C10 + #{index}*32

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-280: cmn_hns_rn_cluster0-127_physid_reg2

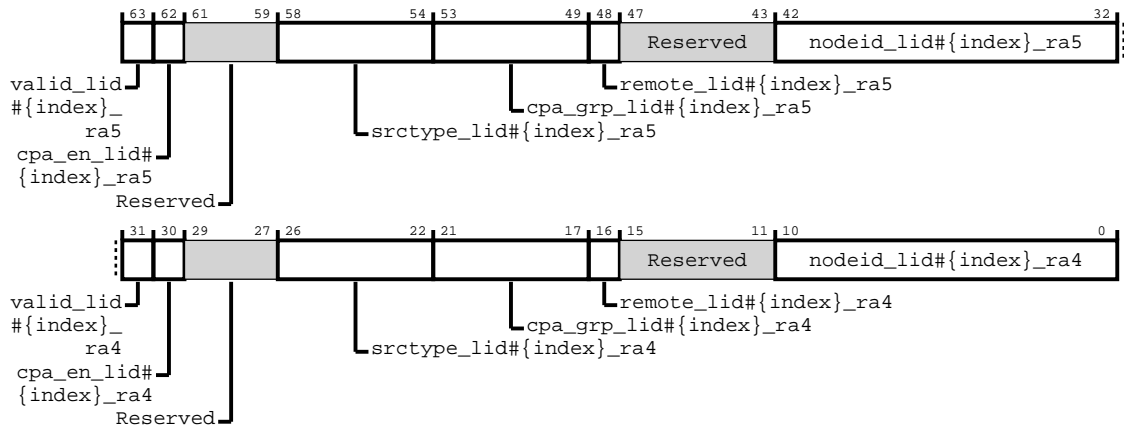


Table 5-284: cmn_hns_rn_cluster0-127_physid_reg2 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra5	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra5	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[61:59]	Reserved	Reserved	RO	-
[58:54]	srctype_lid#{index}_ra5	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[53:49]	cpa_grp_lid#{index}_ra5	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra5	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra5	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra4	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra4	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[29:27]	Reserved	Reserved	RO	-
[26:22]	srctype_lid#{index}_ra4	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[21:17]	cpa_grp_lid#{index}_ra4	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra4	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra4	Specifies the node ID	RW	11'h0

5.2.4.137 cmn_hns_rn_cluster0-127_physid_reg3

There are 128 iterations of this register. The index ranges from 0 to 127. Configures node IDs for RNs in the system corresponding to each RN ID.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h3C18 + #{index}*32

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-281: cmn_hns_rn_cluster0-127_physid_reg3

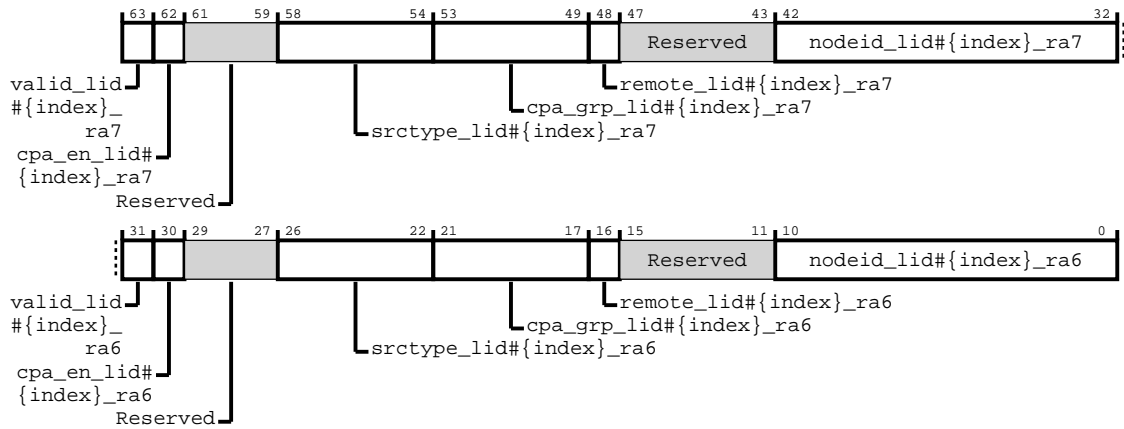


Table 5-285: cmn_hns_rn_cluster0-127_physid_reg3 attributes

Bits	Name	Description	Type	Reset
[63]	valid_lid#{index}_ra7	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[62]	cpa_en_lid#{index}_ra7	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[61:59]	Reserved	Reserved	RO	-
[58:54]	srctype_lid#{index}_ra7	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[53:49]	cpa_grp_lid#{index}_ra7	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[48]	remote_lid#{index}_ra7	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[47:43]	Reserved	Reserved	RO	-
[42:32]	nodeid_lid#{index}_ra7	Specifies the node ID	RW	11'h0
[31]	valid_lid#{index}_ra6	Specifies whether the RN is valid 1'b0: RN ID is not valid 1'b1: RN ID is pointing to a valid CHI device	RW	1'h0
[30]	cpa_en_lid#{index}_ra6	Specifies whether the CCIX port aggregation is enabled 1'b0: CPA not enabled 1'b1: CPA enabled	RW	1'h0
[29:27]	Reserved	Reserved	RO	-
[26:22]	srctype_lid#{index}_ra6	Specifies the CHI source type of the RN 5'b01010: 256 bit CHI-B RN-F 5'b01011: 256 bit CHI-C RN-F 5'b01100: 256 bit CHI-D RN-F 5'b01101: 256 bit CHI-E RN-F 5'b10000: 256 bit CHI-F RN-F Others : Reserved	RW	5'b00000
[21:17]	cpa_grp_lid#{index}_ra6	Specifies CCIX port aggregation group ID(0-31)	RW	5'h0
[16]	remote_lid#{index}_ra6	Specifies whether the RN is remote or local 1'b0: Local RN 1'b1: Remote RN	RW	1'h0
[15:11]	Reserved	Reserved	RO	-
[10:0]	nodeid_lid#{index}_ra6	Specifies the node ID	RW	11'h0

5.2.4.138 cmn_hns_sam_nonhash_cfg1_memregion2-63

There are 62 iterations of this register. The index ranges from 2 to 63. Configures non-hashed memory region #{index} in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h5000 + #{index}*8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-282: cmn_hns_sam_nonhash_cfg1_memregion2-63

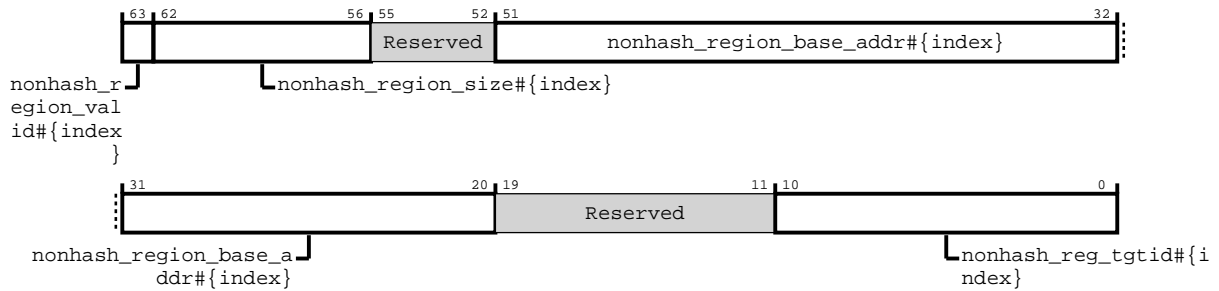


Table 5-286: cmn_hns_sam_nonhash_cfg1_memregion2-63 attributes

Bits	Name	Description	Type	Reset
[63]	<code>nonhash_region_valid#{index}</code>	valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'b0
[62:56]	<code>nonhash_region_size#{index}</code>	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size ($2^{\text{address width}}$).	RW	7'b0
[55:52]	Reserved	Reserved	RO	-
[51:20]	<code>nonhash_region_base_addr#{index}</code>	Bits [51:16] of base address of the range, LSB bit is defined by the parameter <code>POR_HNSAM_RCOMP_LSB_PARAM</code>	RW	32'h0
[19:11]	Reserved	Reserved	RO	-
[10:0]	<code>nonhash_reg_tgtid#{index}</code>	SN TgtID for the non-hashed region	RW	11'h0

5.2.4.139 cmn_hns_sam_nonhash_cfg2_memregion2-63

There are 62 iterations of this register. The index ranges from 2 to 63. Configures non-hashed memory region `#{index}` in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$16'h5200 + \text{#{index}} * 8$

Type

RW

Reset value

See individual bit resets

Root group override

`cmn_hns_rcr.cfg_ctl`

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-283: cmn_hns_sam_nonhash_cfg2_memregion2-63

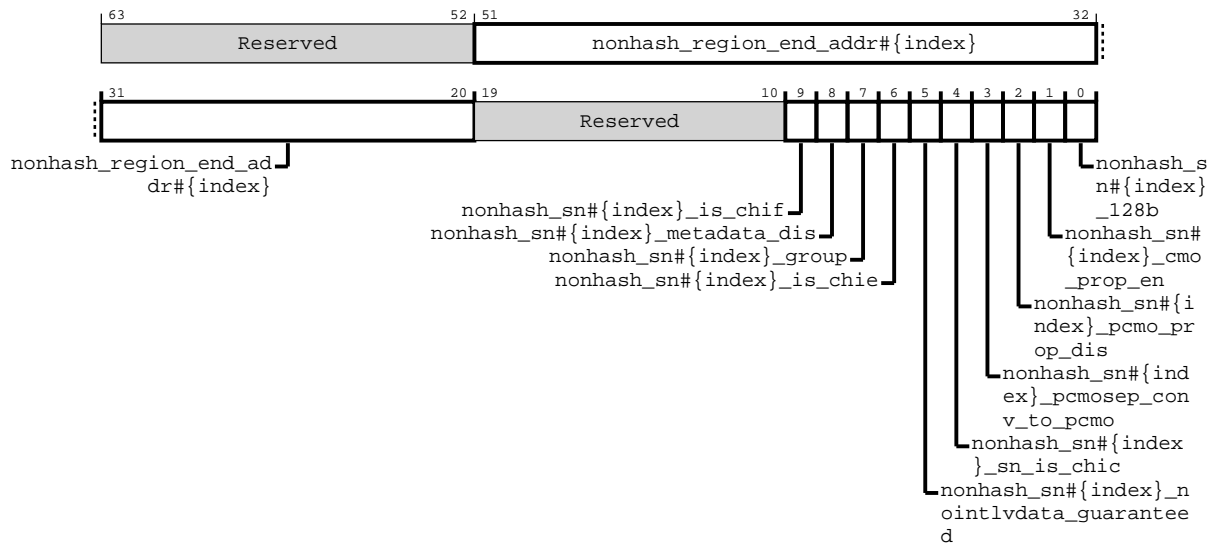


Table 5-287: cmn_hns_sam_nonhash_cfg2_memregion2-63 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	nonhash_region_end_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:10]	Reserved	Reserved	RO	-
[9]	nonhash_sn#{index}_is_chif	nonhash SN #{index} supports CHI-F	RW	1'b0
[8]	nonhash_sn#{index}_metadata_dis	HNS implements metadata termination flow for nonhash SN #{index} when set	RW	1'b0
[7]	nonhash_sn#{index}_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[6]	nonhash_sn#{index}_is_chie	nonhash SN #{index} supports CHI-E	RW	1'b0
[5]	nonhash_sn#{index}_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0

Bits	Name	Description	Type	Reset
[4]	nonhash_sn#{index}_sn_is_chic	Indicates that nonhash sn is a CHI-C SN when set	RW	1'b0
[3]	nonhash_sn#{index}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for nonhash SN #{index} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	nonhash_sn#{index}_pcmo_prop_dis	Disables PCMO propagation for nonhash SN #{index} when set	RW	1'b0
[1]	nonhash_sn#{index}_cmo_prop_en	Enables CMO propagation for nonhash SN #{index} when set	RW	1'b0
[0]	nonhash_sn#{index}_128b	Data width of nonhash SN #{index} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0

5.2.4.140 cmn_hns_sam_htg_cfg1_memregion0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures HTG memory region #{index} in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h5400 + #{index}*8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-284: cmn_hns_sam_htg_cfg1_memregion0-15

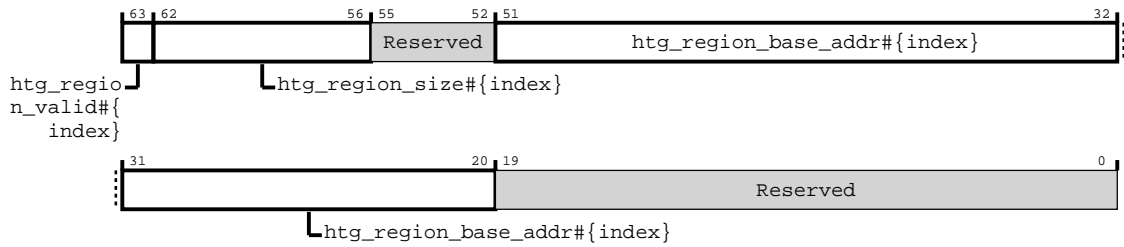


Table 5-288: cmn_hns_sam_htg_cfg1_memregion0-15 attributes

Bits	Name	Description	Type	Reset
[63]	htg_region_valid#{index}	valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'b0
[62:56]	htg_region_size#{index}	Memory region size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b000000
[55:52]	Reserved	Reserved	RO	-
[51:20]	htg_region_base_addr#{index}	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_HNSAM_RCOMP_LSB_PARAM	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

5.2.4.141 cmn_hns_sam_htg_cfg2_memregion0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures htg memory region #{index} in HN-F SAM.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h5480 + #{index}*8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the `cmn_hns_rcr.cfg_ctl` bit is set, Secure accesses to this register are permitted.

If both the `cmn_hns_rcr.cfg_ctl` bit and `cmn_hns_scr.cfg_ctl` bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-285: `cmn_hns_sam_htg_cfg2_memregion0-15`

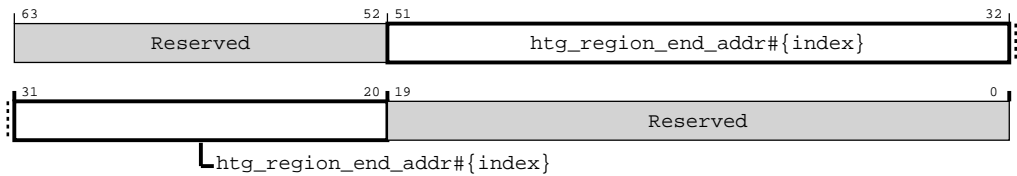


Table 5-289: `cmn_hns_sam_htg_cfg2_memregion0-15` attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:20]	htg_region_end_addr#{index}	Bits [51:20] of base address of the range, LSB bit is defined by the parameter <code>POR_HNSAM_RCOMP_LSB_PARAM</code>	RW	32'h0
[19:0]	Reserved	Reserved	RO	-

5.2.4.142 `cmn_hns_sam_htg_cfg3_memregion0-15`

There are 16 iterations of this register. The index ranges from 0 to 15. Configures the HTG memory region `#{index}`

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$16'h5500 + \#{index} * 8$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-286: cmn_hns_sam_htg_cfg3_memregion0-15

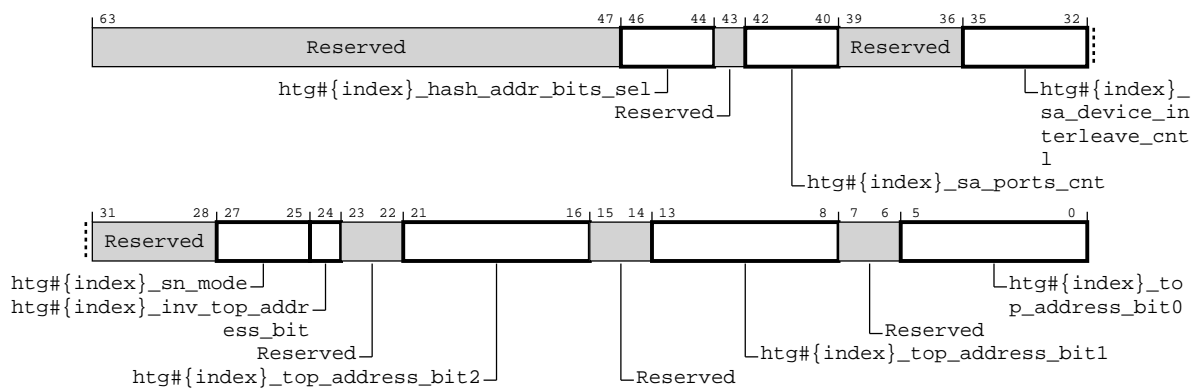


Table 5-290: cmn_hns_sam_htg_cfg3_memregion0-15 attributes

Bits	Name	Description	Type	Reset
[63:47]	Reserved	Reserved	RO	-
[46:44]	htg#{index}_hash_addr_bits_sel	SN hash address select (Valid for 3SN, 5SN, 6SN) 3'b000: [16:8] address bits (Default) 3'b001: [17:9] address bits 3'b010: [18:10] address bits 3'b011: [19:11] address bits 3'b100: [20:12] address bits 3'b101: [21:13] address bits Others: Reserved	RW	3'h0
[43]	Reserved	Reserved	RO	-
[42:40]	htg#{index}_sa_ports_cnt	Specifies the number of CXSA/CXLSA device aggregated 3'b000: 1 port 3'b001: 2 ports 3'b010: 4 ports 3'b011: 8 ports 3'b100: 16 ports 3'b101: 3 ports 3'b110: 6 ports 3'b111: 12 ports	RW	3'b0
[39:36]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[35:32]	htg#{index}_sa_device_interleave_cntl	This field controls the interleave size across all aggregated CXSA/CXLSA Devices 4'h0: 64B Interleaved 4'h1: 128B Interleaved 4'h2: 256B Interleaved 4'h3: 512B Interleaved 4'h4: 1KB Interleaved 4'h5: 2KB Interleaved 4'h6: 4KB Interleaved 4'h7: 8KB Interleaved 4'h8: 16KB Interleaved Others: Reserved	RW	4'b0
[31:28]	Reserved	Reserved	RO	-
[27:25]	htg#{index}_sn_mode	SN selection mode 3'b000: Reserved 3'b001: 3-SN mode (SN0, SN1, SN2) 3'b010: 6-SN mode (SN0, SN1, SN2, SN3, SN4, SN5) 3'b011: 5-SN mode (SN0, SN1, SN2, SN3, SN4) 3'b100: 2-SN mode (SN0, SN1) power of 2 hashing 3'b101: 4-SN mode (SN0, SN1, SN2, SN3) power of 2 hashing 3'b110: 8-SN mode (SN0, SN1, SN2, SN3, SN4, SN5, SN6, SN7) power of 2 hashing 3'b111: CXSA/CXLSA aggregated SA selection function	RW	3'b0
[24]	htg#{index}_inv_top_address_bit	Inverts the top address bit (top_address_bit1 if 3-SN, top_address_bit2 if 6-SN); only used when the address map does not have unique address bit combinations	RW	1'h0
[23:22]	Reserved	Reserved	RO	-
[21:16]	htg#{index}_top_address_bit2	Top address bit 2	RW	6'h00
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg#{index}_top_address_bit1	Top address bit 1	RW	6'h00
[7:6]	Reserved	Reserved	RO	-
[5:0]	htg#{index}_top_address_bit0	Top address bit 0	RW	6'h00

5.2.4.143 cmn_hns_sam_htg_sn_nodeid_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures SN node IDs for HTGs in the HNSAM . Controls target SN node IDs #{index*4 + 0} to #{index*4 + 3}.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h5600 + #{index}*8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-287: cmn_hns_sam_htg_sn_nodeid_reg0-15

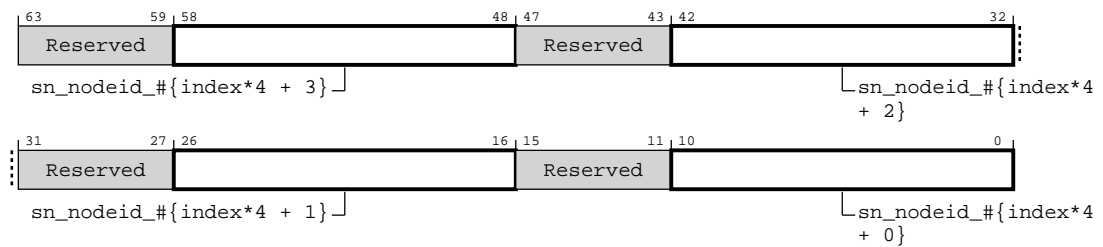


Table 5-291: cmn_hns_sam_htg_sn_nodeid_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	sn_nodeid_{index*4 + 3}	Hashed target SN node ID #{index*4 + 3}	RW	11'b000000000000
[47:43]	Reserved	Reserved	RO	-
[42:32]	sn_nodeid_{index*4 + 2}	Hashed target SN node ID #{index*4 + 2}	RW	11'b000000000000
[31:27]	Reserved	Reserved	RO	-
[26:16]	sn_nodeid_{index*4 + 1}	Hashed target SN node ID #{index*4 + 1}	RW	11'b000000000000
[15:11]	Reserved	Reserved	RO	-
[10:0]	sn_nodeid_{index*4 + 0}	Hashed target SN node ID #{index*4 + 0}	RW	11'b000000000000

5.2.4.144 cmn_hns_sam_htg_sn_attr0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures SN node attributes HTGs in the HNSAM . Controls SN attributes #{index*4 + 0} to #{index*4 + 3}.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$16'h5680 + \#\{index\} * 8$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-288: cmn_hns_sam_htg_sn_attr0-15

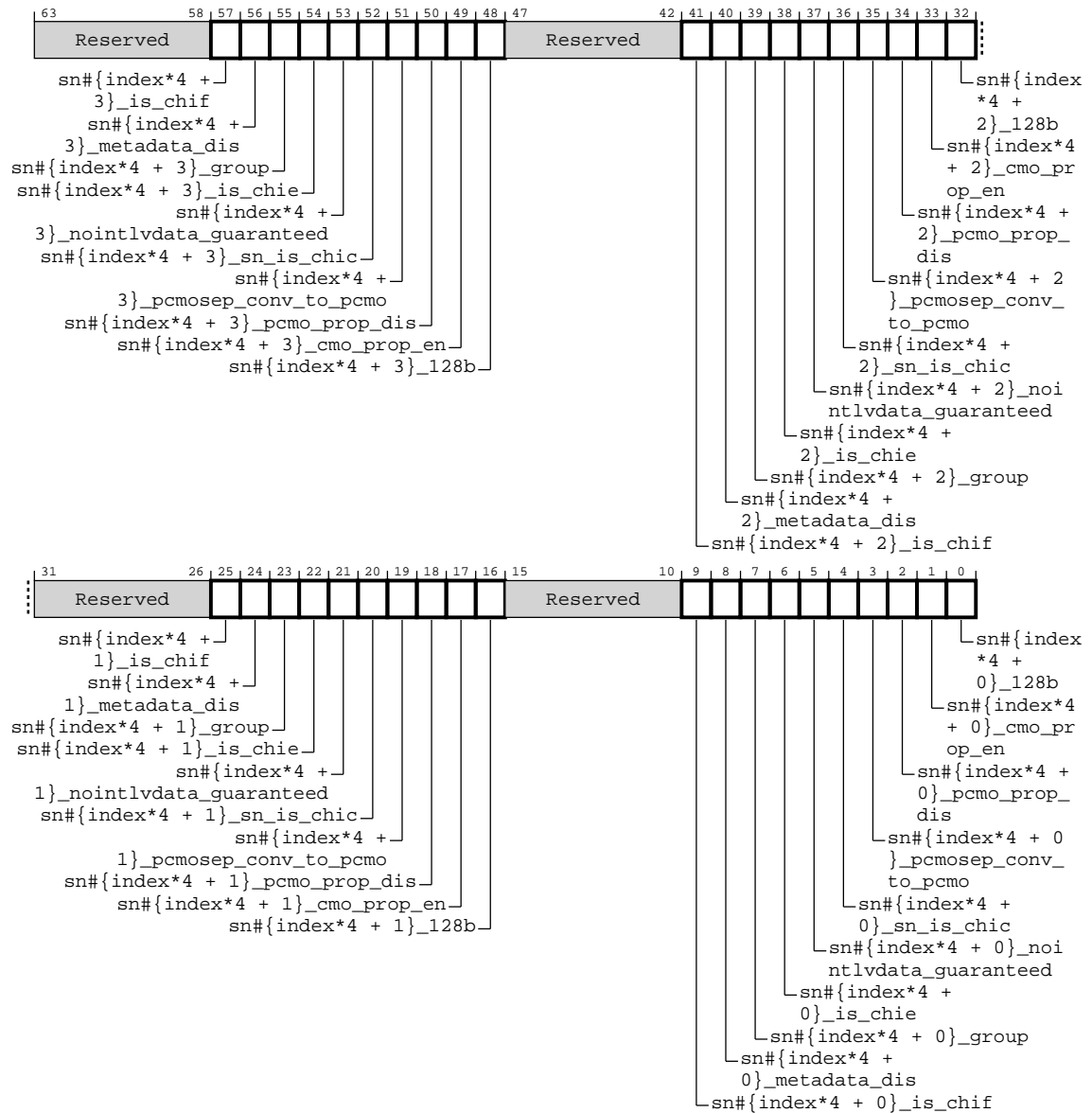


Table 5-292: cmn_hns_sam_htg_sn_attr0-15 attributes

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57]	sn#{index*4 + 3}_is_chif	SN #[index*4 + 3] supports CHI-F	RW	1'b0
[56]	sn#{index*4 + 3}_metadata_dis	HNS implements metadata termination flow for SN #[index*4 + 3] when set	RW	1'b0
[55]	sn#{index*4 + 3}_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[54]	sn#{index*4 + 3}_is_chie	SN #[index*4 + 3] supports CHI-E	RW	1'b0
[53]	sn#{index*4 + 3}_nointlvdata_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0

Bits	Name	Description	Type	Reset
[52]	sn#{index*4 + 3}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[51]	sn#{index*4 + 3}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 3} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[50]	sn#{index*4 + 3}_pcmoprop_dis	Disables PCMO propagation for SN #{index*4 + 3} when set	RW	1'b0
[49]	sn#{index*4 + 3}_cmoprop_en	Enables CMO propagation for SN #{index*4 + 3} when set	RW	1'b0
[48]	sn#{index*4 + 3}_128b	Data width of SN #{index*4 + 3} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[47:42]	Reserved	Reserved	RO	-
[41]	sn#{index*4 + 2}_is_chif	SN #{index*4 + 2} supports CHI-F	RW	1'b0
[40]	sn#{index*4 + 2}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 2} when set	RW	1'b0
[39]	sn#{index*4 + 2}_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[38]	sn#{index*4 + 2}_is_chie	SN #{index*4 + 2} supports CHI-E	RW	1'b0
[37]	sn#{index*4 + 2}_nointlvd_data_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[36]	sn#{index*4 + 2}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[35]	sn#{index*4 + 2}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 2} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[34]	sn#{index*4 + 2}_pcmoprop_dis	Disables PCMO propagation for SN #{index*4 + 2} when set	RW	1'b0
[33]	sn#{index*4 + 2}_cmoprop_en	Enables CMO propagation for SN #{index*4 + 2} when set	RW	1'b0
[32]	sn#{index*4 + 2}_128b	Data width of SN #{index*4 + 2} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[31:26]	Reserved	Reserved	RO	-
[25]	sn#{index*4 + 1}_is_chif	SN #{index*4 + 1} supports CHI-F	RW	1'b0
[24]	sn#{index*4 + 1}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 1} when set	RW	1'b0
[23]	sn#{index*4 + 1}_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[22]	sn#{index*4 + 1}_is_chie	SN #{index*4 + 1} supports CHI-E	RW	1'b0
[21]	sn#{index*4 + 1}_nointlvd_data_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[20]	sn#{index*4 + 1}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[19]	sn#{index*4 + 1}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 1} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[18]	sn#{index*4 + 1}_pcmoprop_dis	Disables PCMO propagation for SN #{index*4 + 1} when set	RW	1'b0
[17]	sn#{index*4 + 1}_cmoprop_en	Enables CMO propagation for SN #{index*4 + 1} when set	RW	1'b0
[16]	sn#{index*4 + 1}_128b	Data width of SN #{index*4 + 1} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[15:10]	Reserved	Reserved	RO	-
[9]	sn#{index*4 + 0}_is_chif	SN #{index*4 + 0} supports CHI-F	RW	1'b0
[8]	sn#{index*4 + 0}_metadata_dis	HNS implements metadata termination flow for SN #{index*4 + 0} when set	RW	1'b0

Bits	Name	Description	Type	Reset
[7]	sn#{index*4 + 0}_group	Specifies the SN-F grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[6]	sn#{index*4 + 0}_is_chie	SN #{index*4 + 0} supports CHI-E	RW	1'b0
[5]	sn#{index*4 + 0}_nointlvd_data_guaranteed	SN guarantees the return data will not be interleaved	RW	1'b0
[4]	sn#{index*4 + 0}_sn_is_chic	Indicates that sn is a CHI-C SN when set	RW	1'b0
[3]	sn#{index*4 + 0}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for SN #{index*4 + 0} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	sn#{index*4 + 0}_pcmo_prop_dis	Disables PCMO propagation for SN #{index*4 + 0} when set	RW	1'b0
[1]	sn#{index*4 + 0}_cmo_prop_en	Enables CMO propagation for SN #{index*4 + 0} when set	RW	1'b0
[0]	sn#{index*4 + 0}_128b	Data width of SN #{index*4 + 0} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0

5.2.4.145 cmn_hns_sam_ccg_sa_nodeid_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures CCG SA node IDs for HTGs in the HNSAM

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h5700 + #{index}*8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-289: cmn_hns_sam_ccg_sa_nodeid_reg0-3

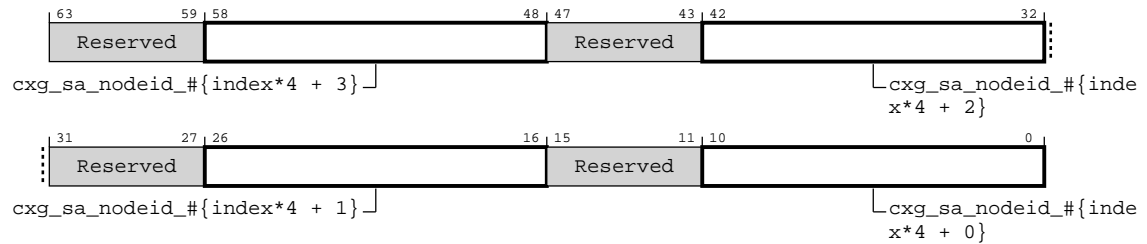


Table 5-293: cmn_hns_sam_ccg_sa_nodeid_reg0-3 attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:48]	<code>cxg_sa_nodeid_#{index*4 + 3}</code>	Hashed target CCG SA node ID <code>#{index*4 + 3}</code>	RW	11'b000000000000
[47:43]	Reserved	Reserved	RO	-
[42:32]	<code>cxg_sa_nodeid_#{index*4 + 2}</code>	Hashed target CCG SA node ID <code>#{index*4 + 2}</code>	RW	11'b000000000000
[31:27]	Reserved	Reserved	RO	-
[26:16]	<code>cxg_sa_nodeid_#{index*4 + 1}</code>	Hashed target CCG SA node ID <code>#{index*4 + 1}</code>	RW	11'b000000000000
[15:11]	Reserved	Reserved	RO	-
[10:0]	<code>cxg_sa_nodeid_#{index*4 + 0}</code>	Hashed target CCG SA node ID <code>#{index*4 + 0}</code>	RW	11'b000000000000

5.2.4.146 cmn_hns_sam_ccg_sa_attr0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Configures CCG SA node attributes.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

`16'h5740 + #{index}*8`

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Bit descriptions

The following image shows the higher register bit assignments.

[illegible]

Bits	Name	Description	Type	Reset
[63:58]	Reserved	Reserved	RO	-
[57]	ccg_sa#{index*4 + 3}_is_chif	CCG_SA #{index*4 + 3} supports CHI-F	RW	1'b0
[56]	ccg_sa#{index*4 + 3}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 3} when set	RW	1'b0

Bits	Name	Description	Type	Reset
[55]	ccg_sa#{index*4 + 3}_group	Specifies the CCG_SA grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[54]	ccg_sa#{index*4 + 3}_is_chie	CCG_SA #{index*4 + 3} supports CHI-E	RW	1'b0
[53]	ccg_sa#{index*4 + 3}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	1'b0
[52]	ccg_sa#{index*4 + 3}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[51]	ccg_sa#{index*4 + 3}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 3} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[50]	ccg_sa#{index*4 + 3}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 3} when set	RW	1'b0
[49]	ccg_sa#{index*4 + 3}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 3} when set	RW	1'b0
[48]	ccg_sa#{index*4 + 3}_128b	Data width of CCG_SA #{index*4 + 3} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[47:42]	Reserved	Reserved	RO	-
[41]	ccg_sa#{index*4 + 2}_is_chif	CCG_SA #{index*4 + 2} supports CHI-F	RW	1'b0
[40]	ccg_sa#{index*4 + 2}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 2} when set	RW	1'b0
[39]	ccg_sa#{index*4 + 2}_group	Specifies the CCG_SA grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[38]	ccg_sa#{index*4 + 2}_is_chie	CCG_SA #{index*4 + 2} supports CHI-E	RW	1'b0
[37]	ccg_sa#{index*4 + 2}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	1'b0
[36]	ccg_sa#{index*4 + 2}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[35]	ccg_sa#{index*4 + 2}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 2} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[34]	ccg_sa#{index*4 + 2}_pcmo_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 2} when set	RW	1'b0
[33]	ccg_sa#{index*4 + 2}_cmo_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 2} when set	RW	1'b0
[32]	ccg_sa#{index*4 + 2}_128b	Data width of CCG_SA #{index*4 + 2} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[31:26]	Reserved	Reserved	RO	-
[25]	ccg_sa#{index*4 + 1}_is_chif	CCG_SA #{index*4 + 1} supports CHI-F	RW	1'b0
[24]	ccg_sa#{index*4 + 1}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 1} when set	RW	1'b0
[23]	ccg_sa#{index*4 + 1}_group	Specifies the CCG_SA grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[22]	ccg_sa#{index*4 + 1}_is_chie	CCG_SA #{index*4 + 1} supports CHI-E	RW	1'b0
[21]	ccg_sa#{index*4 + 1}_nointlvdata_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	1'b0

Bits	Name	Description	Type	Reset
[20]	ccg_sa#{index*4 + 1}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[19]	ccg_sa#{index*4 + 1}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 1} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[18]	ccg_sa#{index*4 + 1}_pcmoe_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 1} when set	RW	1'b0
[17]	ccg_sa#{index*4 + 1}_cmoe_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 1} when set	RW	1'b0
[16]	ccg_sa#{index*4 + 1}_128b	Data width of CCG_SA #{index*4 + 1} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0
[15:10]	Reserved	Reserved	RO	-
[9]	ccg_sa#{index*4 + 0}_is_chif	CCG_SA #{index*4 + 0} supports CHI-F	RW	1'b0
[8]	ccg_sa#{index*4 + 0}_metadata_dis	HNS implements metadata termination flow for CCG_SA #{index*4 + 0} when set	RW	1'b0
[7]	ccg_sa#{index*4 + 0}_group	Specifies the CCG_SA grouping 1'b0: Group A 1'b1: Group B	RW	1'b0
[6]	ccg_sa#{index*4 + 0}_is_chie	CCG_SA #{index*4 + 0} supports CHI-E	RW	1'b0
[5]	ccg_sa#{index*4 + 0}_nointlvdta_guaranteed	CCG_SA guarantees the return data will not be interleaved	RW	1'b0
[4]	ccg_sa#{index*4 + 0}_sn_is_chic	Indicates that CCG_SA is a CHI-C CCG_SA when set	RW	1'b0
[3]	ccg_sa#{index*4 + 0}_pcmosep_conv_to_pcmo	Convert CleanSharedPersistSep to CleanSharedPersist for CCG_SA #{index*4 + 0} when set CONSTRAINT: Should not be enabled when sn_pcmo_prop_dis bit is set to 1	RW	1'b0
[2]	ccg_sa#{index*4 + 0}_pcmoe_prop_dis	Disables PCMO propagation for CCG_SA #{index*4 + 0} when set	RW	1'b0
[1]	ccg_sa#{index*4 + 0}_cmoe_prop_en	Enables CMO propagation for CCG_SA #{index*4 + 0} when set	RW	1'b0
[0]	ccg_sa#{index*4 + 0}_128b	Data width of CCG_SA #{index*4 + 0} 1'b1: 128 bits 1'b0: 256 bits	RW	1'b0

5.2.4.147 hns_generic_regs0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configuration register for the custom logic.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

16'h5780 + #{index}*8

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.cfg_ctl

Secure group override

cmn_hns_scr.cfg_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.cfg_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.cfg_ctl bit and cmn_hns_scr.cfg_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-291: hns_generic_regs0-7

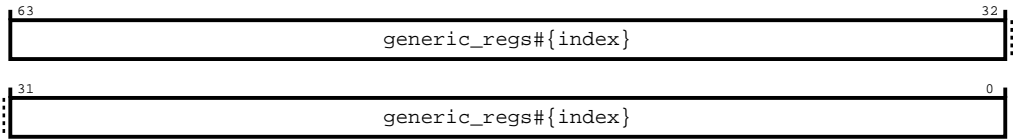


Table 5-295: hns_generic_regs0-7 attributes

Bits	Name	Description	Type	Reset
[63:0]	generic_regs#{index}	Configuration register for the custom logic	RW	64'h0

5.2.4.148 cmn_hns_pa2setaddr_slc

Functions as the control register of PA to SetAddr and vice versa conversion for HNS-SLC

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h5900

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.pa2setaddr_ctl

Secure group override

cmn_hns_scr.pa2setaddr_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.pa2setaddr_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.pa2setaddr_ctl bit and cmn_hns_scr.pa2setaddr_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-292: cmn_hns_pa2setaddr_slc

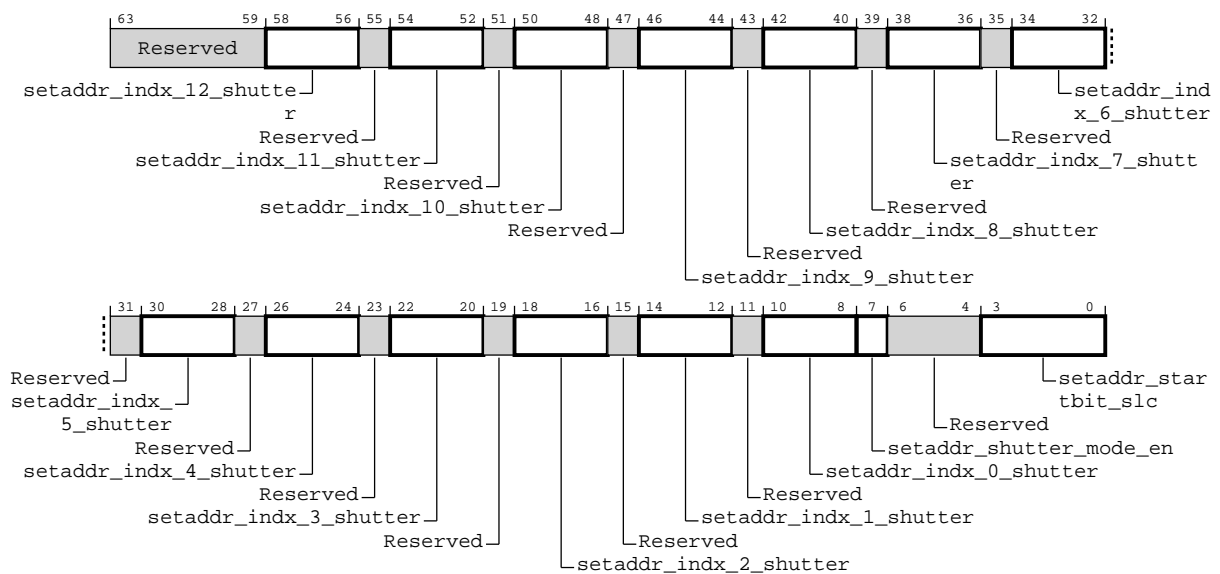


Table 5-296: cmn_hns_pa2setaddr_slc attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:56]	setaddr_idx_12_shutter	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_idx_11_shutter	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_idx_10_shutter	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_idx_9_shutter	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_idx_8_shutter	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_idx_7_shutter	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[35]	Reserved	Reserved	RO	-
[34:32]	setaddr_idx_6_shutter	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_idx_5_shutter	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_idx_4_shutter	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_idx_3_shutter	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[19]	Reserved	Reserved	RO	-
[18:16]	setaddr_idx_2_shutter	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[15]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[14:12]	setaddr_indx_1_shutter	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[11]	Reserved	Reserved	RO	-
[10:8]	setaddr_indx_0_shutter	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_slc 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[7]	setaddr_shutter_mode_en	Enables address shuttering mode for SLC as programmed by setaddr_indx_X_shutter registers	RW	1'b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_slc	SLC: SetAddr starting bit for SLC TODO add a description here abt contiguous bits 4'b0110: Setaddr starts from PA[6] 4'b0111: Setaddr starts from PA[7] 4'b1000: Setaddr starts from PA[8] 4'b1001: Setaddr starts from PA[9] 4'b1010: Setaddr starts from PA[10] 4'b1011: Setaddr starts from PA[11] 4'b1100: Setaddr starts from PA[12]	RW	4'b0110

5.2.4.149 cmn_hns_pa2setaddr_sf

Functions as the control register of PA to Set/TagAddr and vice versa conversion for HNS-SF

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h5908

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.pa2setaddr_ctl

Secure group override

cmn_hns_scr.pa2setaddr_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.pa2setaddr_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.pa2setaddr_ctl bit and cmn_hns_scr.pa2setaddr_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-293: cmn_hns_pa2setaddr_sf

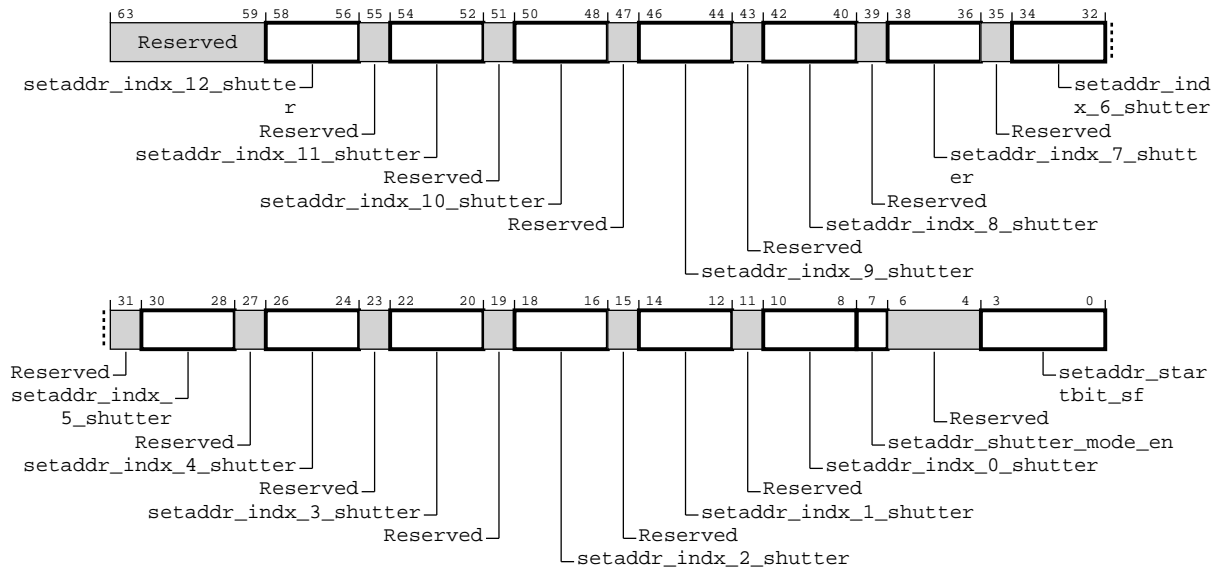


Table 5-297: cmn_hns_pa2setaddr_sf attributes

Bits	Name	Description	Type	Reset
[63:59]	Reserved	Reserved	RO	-
[58:56]	setaddr_indx_12_shutter	Program to specify address bit shuttering for setaddr index 12 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[55]	Reserved	Reserved	RO	-
[54:52]	setaddr_indx_11_shutter	Program to specify address bit shuttering for setaddr index 11 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[51]	Reserved	Reserved	RO	-
[50:48]	setaddr_indx_10_shutter	Program to specify address bit shuttering for setaddr index 10 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[47]	Reserved	Reserved	RO	-
[46:44]	setaddr_indx_9_shutter	Program to specify address bit shuttering for setaddr index 9 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[43]	Reserved	Reserved	RO	-
[42:40]	setaddr_indx_8_shutter	Program to specify address bit shuttering for setaddr index 8 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0

Bits	Name	Description	Type	Reset
[39]	Reserved	Reserved	RO	-
[38:36]	setaddr_indx_7_shutter	Program to specify address bit shuttering for setaddr index 7 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[35]	Reserved	Reserved	RO	-
[34:32]	setaddr_indx_6_shutter	Program to specify address bit shuttering for setaddr index 6 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[31]	Reserved	Reserved	RO	-
[30:28]	setaddr_indx_5_shutter	Program to specify address bit shuttering for setaddr index 5 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[27]	Reserved	Reserved	RO	-
[26:24]	setaddr_indx_4_shutter	Program to specify address bit shuttering for setaddr index 4 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[23]	Reserved	Reserved	RO	-
[22:20]	setaddr_indx_3_shutter	Program to specify address bit shuttering for setaddr index 3 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[19]	Reserved	Reserved	RO	-
[18:16]	setaddr_indx_2_shutter	Program to specify address bit shuttering for setaddr index 2 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[15]	Reserved	Reserved	RO	-
[14:12]	setaddr_indx_1_shutter	Program to specify address bit shuttering for setaddr index 1 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[11]	Reserved	Reserved	RO	-
[10:8]	setaddr_indx_0_shutter	Program to specify address bit shuttering for setaddr index 0 from the setaddr_startbit_sf 3'b000 : pass-through 3'b001 : shift_1 3'b010 : shift_2 3'b011 : shift_3 3'b100 : shift_4 3'b101 : shift_5	RW	3'b0
[7]	setaddr_shutter_mode_en	Enables address shuttering mode for SF as programmed by setaddr_indx_X_shutter registers	RW	1'b0
[6:4]	Reserved	Reserved	RO	-
[3:0]	setaddr_startbit_sf	SF: SetAddr starting bit for SF 4'b0110: Setaddr starts from PA[6] 4'b0111: Setaddr starts from PA[7] 4'b1000: Setaddr starts from PA[8] 4'b1001: Setaddr starts from PA[9] 4'b1010: Setaddr starts from PA[10] 4'b1011: Setaddr starts from PA[11] 4'b1100: Setaddr starts from PA[12]	RW	4'b0110

5.2.4.150 cmn_hns_pa2setaddr_flex_slc

Functions as the SLC control register of PA to Set/TagAddr and vice versa conversion for HNS (flexible)

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h5910

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.pa2setaddr_ctl

Secure group override

cmn_hns_scr.pa2setaddr_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.pa2setaddr_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.pa2setaddr_ctl bit and cmn_hns_scr.pa2setaddr_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-294: cmn_hns_pa2setaddr_flex_slc

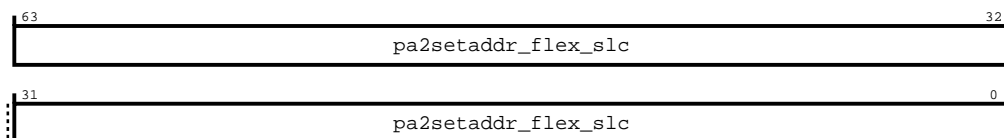


Table 5-298: cmn_hns_pa2setaddr_flex_slc attributes

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_slc	FLEXIBLE: PA to SET/TAG ADDR and vice versa conversion config field for SLC	RW	64'b0

5.2.4.151 cmn_hns_pa2setaddr_flex_sf

Functions as the SF control register of PA to Set/TagAddr and vice versa conversion for HNS (flexible)

Configurations

This register is available in all configurations.

Attributes

Width

64

Address offset

16'h5918

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.pa2setaddr_ctl

Secure group override

cmn_hns_scr.pa2setaddr_ctl

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.pa2setaddr_ctl bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.pa2setaddr_ctl bit and cmn_hns_scr.pa2setaddr_ctl bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-295: cmn_hns_pa2setaddr_flex_sf

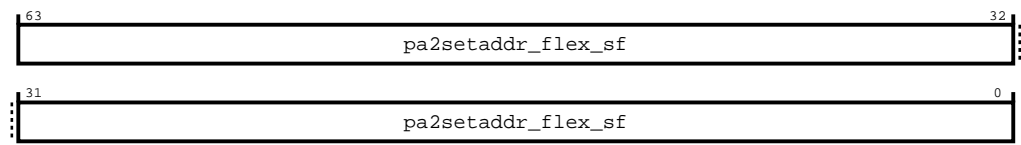


Table 5-299: cmn_hns_pa2setaddr_flex_sf attributes

Bits	Name	Description	Type	Reset
[63:0]	pa2setaddr_flex_sf	FLEXIBLE: PA to SET/TAG ADDR conversion and vice versa config field for SF	RW	64'b0

5.2.4.152 lcn_hashed_tgt_grp_cfg1_region0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

0xindex(0-31) : 16'h7000 + #{8 * index}

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-296: lcn_hashed_tgt_grp_cfg1_region0-31

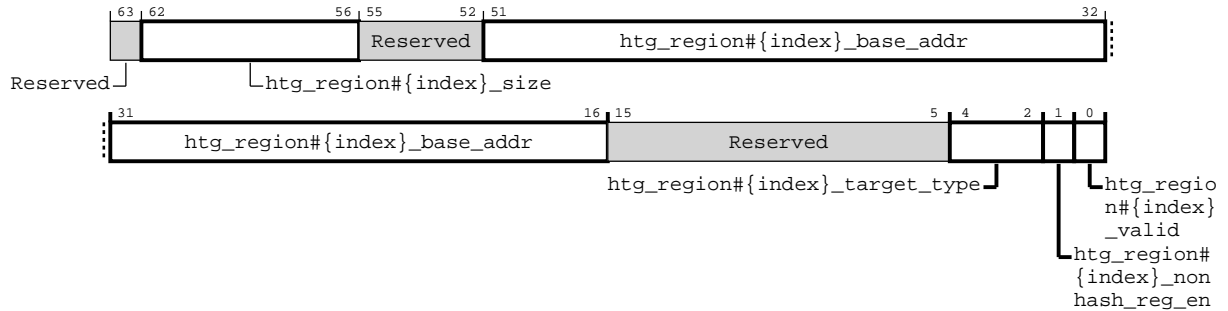


Table 5-300: lcn_hashed_tgt_grp_cfg1_region0-31 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_region#{index}_size	Memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	7'b0000000
[55:52]	Reserved	Reserved	RO	-
[51:16]	htg_region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:5]	Reserved	Reserved	RO	-
[4:2]	htg_region#{index}_target_type	Indicates node type 3'b000: HN-F 3'b001: HN-I 3'b010: CXRA 3'b011: HN-P 3'b100: PCI-CXRA 3'b101: HN-S Others: Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	htg_region#{index}_nonhash_reg_en	Enables hashed region #{index} to select non-hashed node	RW	1'b0
[0]	htg_region#{index}_valid	Memory region #{index} valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'b0

5.2.4.153 lcn_hashed_tgt_grp_cfg2_region0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-31) : 16'h7100 + \#{8 * index}$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-297: lcn_hashed_tgt_grp_cfg2_region0-31

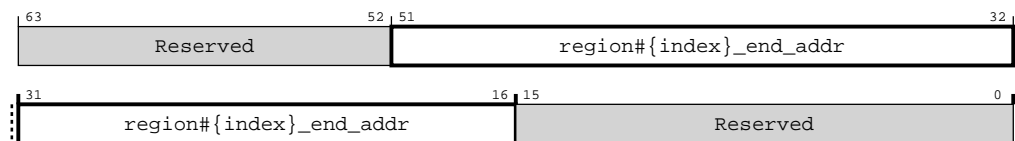


Table 5-301: lcn_hashed_tgt_grp_cfg2_region0-31 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	region#{index}_end_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:0]	Reserved	Reserved	RO	-

5.2.4.154 lcn_hashed_target_grp_secondary_cfg1_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures secondary hashed memory regions

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0x\text{index}(0-31) : 16'h7200 + \{8 * \text{index}\}$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-298: lcn_hashed_target_grp_secondary_cfg1_reg0-31

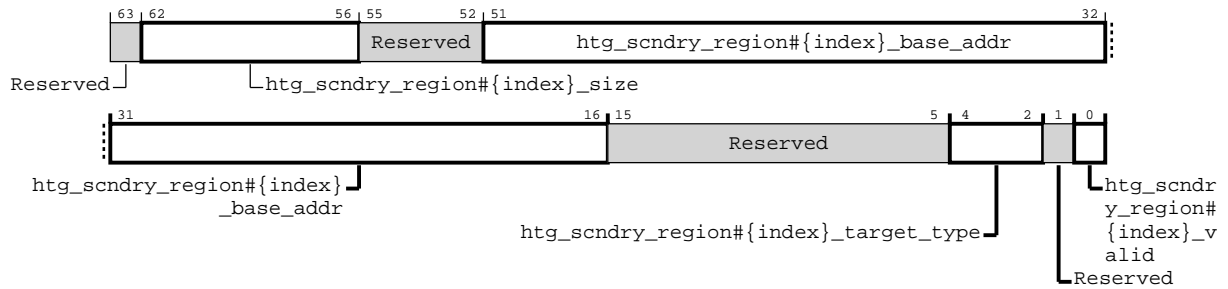


Table 5-302: lcn_hashed_target_grp_secondary_cfg1_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63]	Reserved	Reserved	RO	-
[62:56]	htg_scndry_region#{index}_size	Secondary memory region #{index} size CONSTRAINT: Memory region must be a power of two, from minimum size supported to maximum memory size (2^address width).	RW	5'b00000
[55:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_base_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'h0
[15:5]	Reserved	Reserved	RO	-
[4:2]	htg_scndry_region#{index}_target_type	Indicates node type 3'b000: HN-F 3'b001: HN-I 3'b010: CXRA 3'b011: HN-P 3'b100: PCI-CXRA 3'b101: HN-S Others: Reserved CONSTRAINT: Only applicable for RN-I	RW	3'b000
[1]	Reserved	Reserved	RO	-
[0]	htg_scndry_region#{index}_valid	Secondary memory region #{index} valid 1'b0: Not valid 1'b1: Valid for memory region comparison	RW	1'b0

5.2.4.155 lcn_hashed_target_grp_secondary_cfg2_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures hashed memory regions

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

0xindex(0-31) : 16'h7300 + #{8 * index}

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-299: lcn_hashed_target_grp_secondary_cfg2_reg0-31

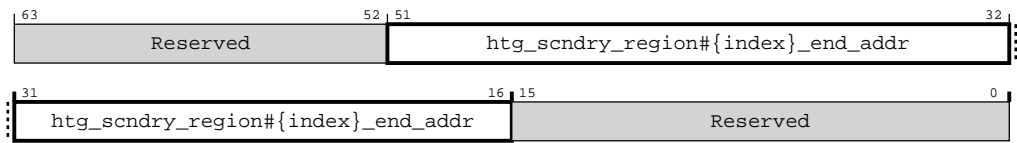


Table 5-303: lcn_hashed_target_grp_secondary_cfg2_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51:16]	htg_scndry_region#{index}_end_addr	Bits [51:16] of base address of the range, LSB bit is defined by the parameter POR_RNSAM_HTG_RCOMP_LSB_PARAM	RW	36'b00000000000000000000000000000000
[15:0]	Reserved	Reserved	RO	-

5.2.4.156 lcn_hashed_target_grp_hash_cntl_reg0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures HTG hash type

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-31) : 16'h7400 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-300: lcn_hashed_target_grp_hash_cntl_reg0-31

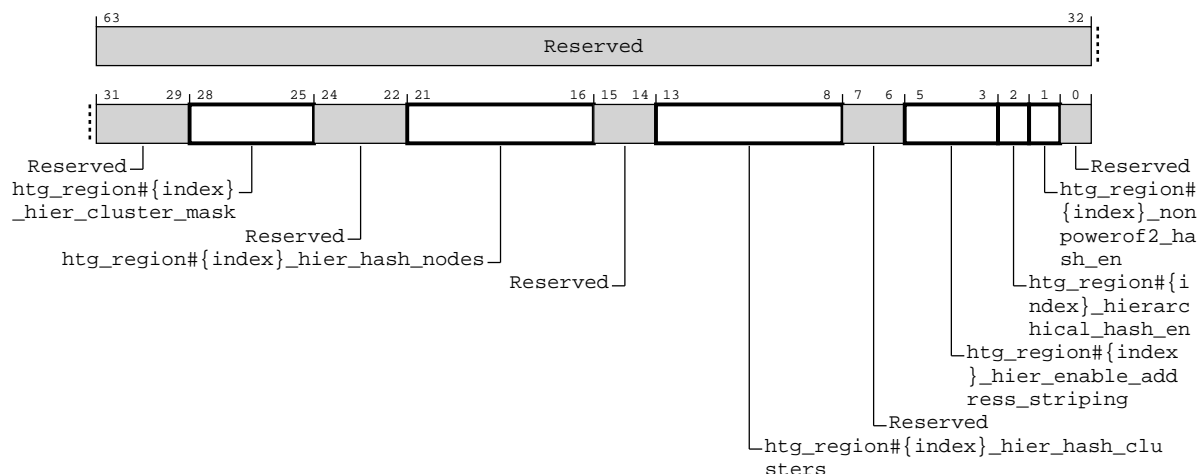


Table 5-304: lcn_hashed_target_grp_hash_cntl_reg0-31 attributes

Bits	Name	Description	Type	Reset
[63:29]	Reserved	Reserved	RO	-
[28:25]	htg_region#{index}_hier_cluster_mask	Hierarchical hashing: Enable cluster masking to achieve different interleave granularity across clusters. 4'b0000: 64 byte interleave granularity across clusters 4'b0001: 128 byte interleave granularity across clusters 4'b0010: 256 byte interleave granularity across clusters 4'b0011: 512 byte interleave granularity across clusters 4'b0100: 1024 byte interleave granularity across clusters 4'b0101: 2048 byte interleave granularity across clusters 4'b0110: 4096 byte interleave granularity across clusters 4'b0111: 8192 byte interleave granularity across clusters others: Reserved	RW	4'b0
[24:22]	Reserved	Reserved	RO	-
[21:16]	htg_region#{index}_hier_hash_nodes	Hierarchical hashing mode, define number of nodes in each cluster	RW	6'h0
[15:14]	Reserved	Reserved	RO	-
[13:8]	htg_region#{index}_hier_hash_clusters	Hierarchical hashing mode, define number of clusters groups	RW	6'h0
[7:6]	Reserved	Reserved	RO	-
[5:3]	htg_region#{index}_hier_enable_address_stripping	Hierarchical hashing: configure number of address bits needs to shuttered (removed) at second hierarchy hash (LSB bit is based on cluster mask). 3'b000: no address shuttering 3'b001: one addr bit shuttered (2 clusters) 3'b010: two addr bit shuttered (4 clusters) 3'b011: three addr bit shuttered (8 clusters) 3'b100: four addr bit shuttered (16 clusters) 3'b101: five addr bit shuttered (32 clusters) others: Reserved	RW	3'b0
[2]	htg_region#{index}_hierarchical_hash_en	Hierarchical Hashing mode enable configure bit	RW	1'b0
[1]	htg_region#{index}_nonpowerof2_hash_en	Non power of two Hashing mode enable cconfigure bit	RW	1'b0
[0]	Reserved	Reserved	RO	-

5.2.4.157 lcn_hashed_target_group_hn_count_reg0-3

There are 4 iterations of this register. The index ranges from 0 to 3. Indicates number of HN-F/HN-P's in hashed target groups #{index*8} to #{index*8 + 7}.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

0xindex(0-3) : 16'h7500 + #{8 * index}

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-301: lcn_hashed_target_group_hn_count_reg0-3

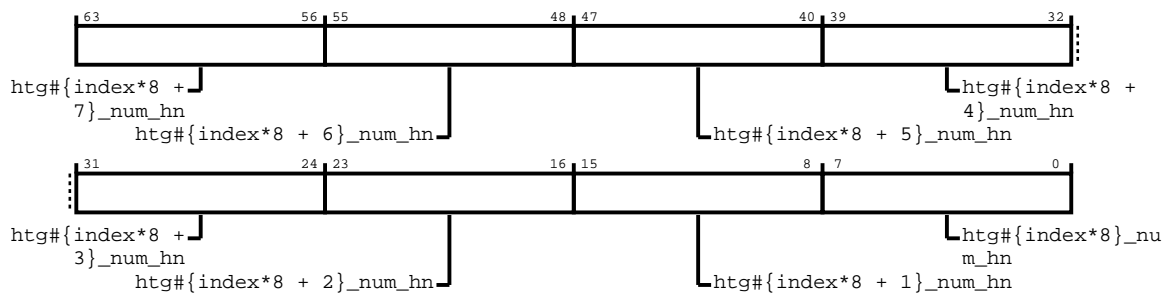


Table 5-305: lcn_hashed_target_group_hn_count_reg0-3 attributes

Bits	Name	Description	Type	Reset
[63:56]	htg#{index*8 + 7}_num_hn	HN count for hashed target group 7	RW	8'h00
[55:48]	htg#{index*8 + 6}_num_hn	HN count for hashed target group 6	RW	8'h00
[47:40]	htg#{index*8 + 5}_num_hn	HN count for hashed target group 5	RW	8'h00
[39:32]	htg#{index*8 + 4}_num_hn	HN count for hashed target group 4	RW	8'h00
[31:24]	htg#{index*8 + 3}_num_hn	HN count for hashed target group 3	RW	8'h00
[23:16]	htg#{index*8 + 2}_num_hn	HN count for hashed target group 2	RW	8'h00
[15:8]	htg#{index*8 + 1}_num_hn	HN count for hashed target group 1	RW	8'h00
[7:0]	htg#{index*8}_num_hn	HN count for hashed target group 0	RW	8'h00

5.2.4.158 lcn_hashed_target_grp_cal_mode_reg0-7

There are 8 iterations of this register. The index ranges from 0 to 7. Configures the HN CAL mode support for all hashed target groups.

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-7) : 16'h7520 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Root group override

cmn_hns_rcr.sam_control

Secure group override

cmn_hns_scr.sam_control

Usage constraints

This register is owned in the Root space and accessible using Root transactions, unless the following override options are set:

If the cmn_hns_rcr.sam_control bit is set, Secure accesses to this register are permitted.

If both the cmn_hns_rcr.sam_control bit and cmn_hns_scr.sam_control bit are set, Non-secure and Realm accesses to this register are permitted.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-302: lcn_hashed_target_grp_cal_mode_reg0-7

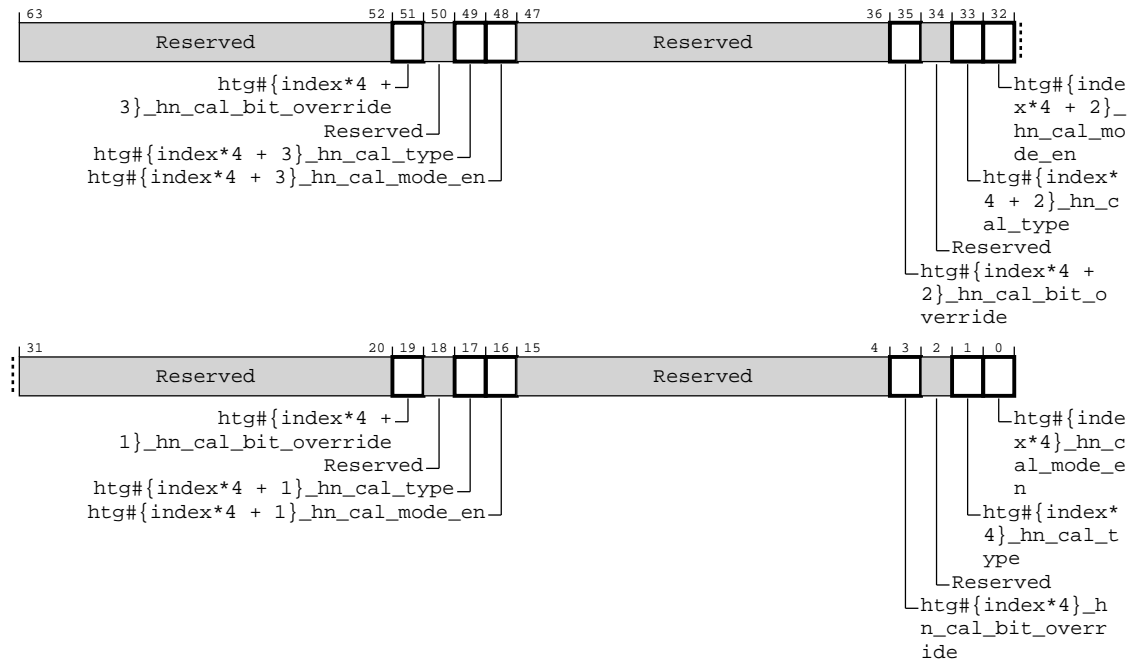


Table 5-306: lcn_hashed_target_grp_cal_mode_reg0-7 attributes

Bits	Name	Description	Type	Reset
[63:52]	Reserved	Reserved	RO	-
[51]	<code>htg#{index*4 + 3}_hn_cal_bit_override</code>	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 3} 1'b0: Hash MSB bit to override Device ID 1'b1: Hash LSB bit to override Device ID	RW	1'b0
[50]	Reserved	Reserved	RO	-
[49]	<code>htg#{index*4 + 3}_hn_cal_type</code>	Enables type of HN CAL for HTG #{index*4 + 3} 1'b0: CAL2 mode 1'b1: CAL4 mode	RW	1'b0
[48]	<code>htg#{index*4 + 3}_hn_cal_mode_en</code>	Enables support for HN CAL for HTG #{index*4 + 3}	RW	1'b0
[47:36]	Reserved	Reserved	RO	-
[35]	<code>htg#{index*4 + 2}_hn_cal_bit_override</code>	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 2} 1'b0: Hash MSB bit to override Device ID 1'b1: Hash LSB bit to override Device ID	RW	1'b0
[34]	Reserved	Reserved	RO	-
[33]	<code>htg#{index*4 + 2}_hn_cal_type</code>	Enables type of HN CAL for HTG #{index*4 + 2} 1'b0: CAL2 mode 1'b1: CAL4 mode	RW	1'b0
[32]	<code>htg#{index*4 + 2}_hn_cal_mode_en</code>	Enables support for HN CAL for HTG #{index*4 + 2}	RW	1'b0
[31:20]	Reserved	Reserved	RO	-
[19]	<code>htg#{index*4 + 1}_hn_cal_bit_override</code>	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4 + 1} 1'b0: Hash MSB bit to override Device ID 1'b1: Hash LSB bit to override Device ID	RW	1'b0
[18]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[17]	htg#{index*4 + 1}_hn_cal_type	Enables type of HN CAL for HTG #{index*4 + 1} 1'b0: CAL2 mode 1'b1: CAL4 mode	RW	1'b0
[16]	htg#{index*4 + 1}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4 + 1}	RW	1'b0
[15:4]	Reserved	Reserved	RO	-
[3]	htg#{index*4}_hn_cal_bit_override	Configuration to choose LSB/MSB bit to override Device ID for HTG #{index*4} 1'b0: Hash MSB bit to override Device ID 1'b1: Hash LSB bit to override Device ID	RW	1'b0
[2]	Reserved	Reserved	RO	-
[1]	htg#{index*4}_hn_cal_type	Enables type of HN CAL for HTG #{index*4} 1'b0: CAL2 mode 1'b1: CAL4 mode	RW	1'b0
[0]	htg#{index*4}_hn_cal_mode_en	Enables support for HN CAL for HTG #{index*4}	RW	1'b0

5.2.4.159 lcn_hashed_target_grp_hnf_cpa_en_reg0-1

There are 2 iterations of this register. The index ranges from 0 to 1. Configures CCIX port aggregation mode for hashed HNF node IDs

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-1) : 16'h7560 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-303: lcn_hashed_target_grp_hnf_cpa_en_reg0-1

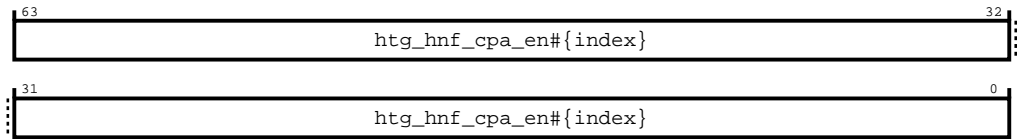


Table 5-307: lcn_hashed_target_grp_hnf_cpa_en_reg0-1 attributes

Bits	Name	Description	Type	Reset
[63:0]	htg_hnf_cpa_en#{index}	Enable CPA for each hashed HNF node ID	RW	64'h0000000000000000

5.2.4.160 lcn_hashed_target_grp_cpag_perhnf_reg0-15

There are 16 iterations of this register. The index ranges from 0 to 15. Configures CPAG ID for each hashed HNF node IDs

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

`0xindex(0-15) : 16'h7580 + #{8 * index}`

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.
Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-304: lcn_hashed_target_grp_cpag_perhnf_reg0-15

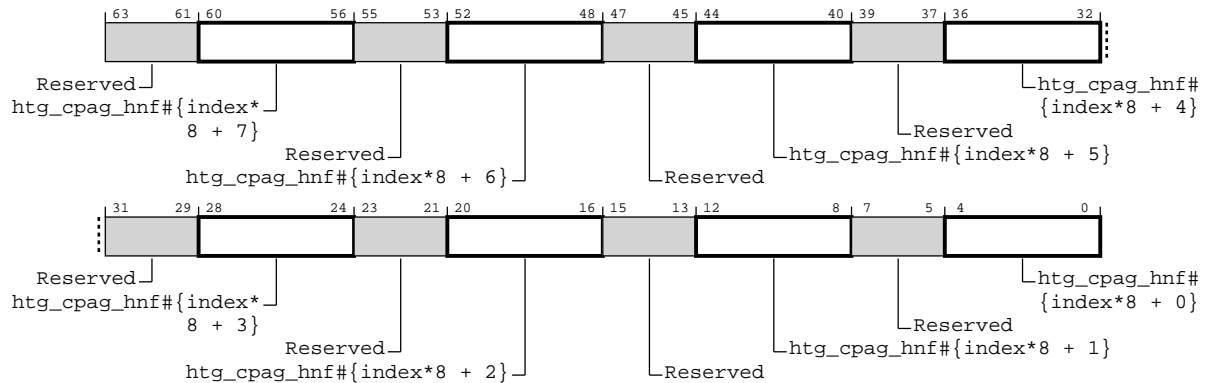


Table 5-308: lcn_hashed_target_grp_cpag_perhnf_reg0-15 attributes

Bits	Name	Description	Type	Reset
[63:61]	Reserved	Reserved	RO	-
[60:56]	htg_cpag_hnf#{index*8 + 7}	CPAG associated to the HNF#{index*8 + 7}	RW	5'b0
[55:53]	Reserved	Reserved	RO	-
[52:48]	htg_cpag_hnf#{index*8 + 6}	CPAG associated to the HNF#{index*8 + 6}	RW	5'b0
[47:45]	Reserved	Reserved	RO	-
[44:40]	htg_cpag_hnf#{index*8 + 5}	CPAG associated to the HNF#{index*8 + 5}	RW	5'b0
[39:37]	Reserved	Reserved	RO	-
[36:32]	htg_cpag_hnf#{index*8 + 4}	CPAG associated to the HNF#{index*8 + 4}	RW	5'b0
[31:29]	Reserved	Reserved	RO	-
[28:24]	htg_cpag_hnf#{index*8 + 3}	CPAG associated to the HNF#{index*8 + 3}	RW	5'b0
[23:21]	Reserved	Reserved	RO	-
[20:16]	htg_cpag_hnf#{index*8 + 2}	CPAG associated to the HNF#{index*8 + 2}	RW	5'b0
[15:13]	Reserved	Reserved	RO	-
[12:8]	htg_cpag_hnf#{index*8 + 1}	CPAG associated to the HNF#{index*8 + 1}	RW	5'b0
[7:5]	Reserved	Reserved	RO	-
[4:0]	htg_cpag_hnf#{index*8 + 0}	CPAG associated to the HNF#{index*8 + 0}	RW	5'b0

5.2.4.161 lcn_hashed_target_grp_compact_cpag_ctrl0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the CPAG control for HTG#{index} valid only when `POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1`

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-31) : 16'h7700 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-305: lcn_hashed_target_grp_compact_cpag_ctrl0-31

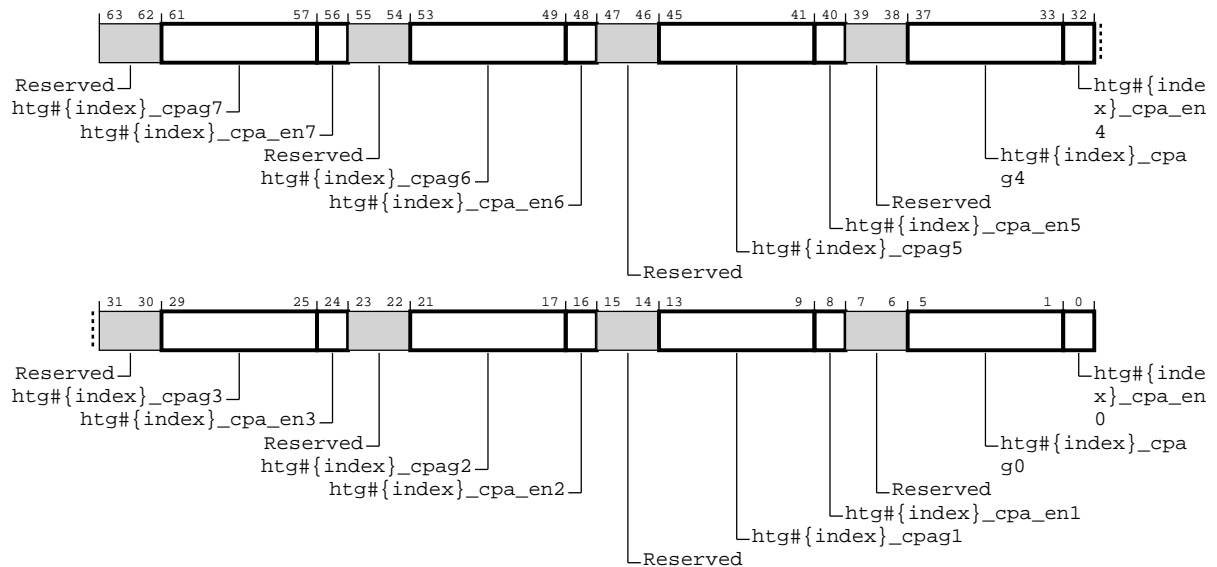


Table 5-309: lcn_hashed_target_grp_compact_cpag_ctrl0-31 attributes

Bits	Name	Description	Type	Reset
[63:62]	Reserved	Reserved	RO	-
[61:57]	htg#{index}_cpag7	cpag id for index7	RW	5'b0
[56]	htg#{index}_cpa_en7	cpa enable for index7	RW	1'b0
[55:54]	Reserved	Reserved	RO	-

Bits	Name	Description	Type	Reset
[53:49]	htg#{index}_cpag6	cpag id for index6	RW	5'b0
[48]	htg#{index}_cpa_en6	cpa enable for index6	RW	1'b0
[47:46]	Reserved	Reserved	RO	-
[45:41]	htg#{index}_cpag5	cpag id for index5	RW	5'b0
[40]	htg#{index}_cpa_en5	cpa enable for index5	RW	1'b0
[39:38]	Reserved	Reserved	RO	-
[37:33]	htg#{index}_cpag4	cpag id for index4	RW	5'b0
[32]	htg#{index}_cpa_en4	cpa enable for index4	RW	1'b0
[31:30]	Reserved	Reserved	RO	-
[29:25]	htg#{index}_cpag3	cpag id for index0	RW	5'b0
[24]	htg#{index}_cpa_en3	cpa enable for index3	RW	1'b0
[23:22]	Reserved	Reserved	RO	-
[21:17]	htg#{index}_cpag2	cpag id for index2	RW	5'b0
[16]	htg#{index}_cpa_en2	cpa enable for index2	RW	1'b0
[15:14]	Reserved	Reserved	RO	-
[13:9]	htg#{index}_cpag1	cpag id for index1	RW	5'b0
[8]	htg#{index}_cpa_en1	cpa enable for index1	RW	1'b0
[7:6]	Reserved	Reserved	RO	-
[5:1]	htg#{index}_cpag0	cpag id for index0	RW	5'b0
[0]	htg#{index}_cpa_en0	cpa enable for index0	RW	1'b0

5.2.4.162 lcn_hashed_target_grp_compact_hash_ctrl0-31

There are 32 iterations of this register. The index ranges from 0 to 31. Configures the HNF hash selection and CPAG hash selection control information for HTG#{index} valid only when POR_RNSAM_COMPACT_HN_TABLES_EN_PARAM == 1

Configurations

This register is only available in specific configurations. In some configurations this register is not present and the bits are reserved.

Attributes

Width

64

Address offset

$0xindex(0-31) : 16'h7800 + \{8 * index\}$

Type

RW

Reset value

See individual bit resets

Usage constraints

This register is owned in the Root space and accessible using Root transactions.

Writes to this register must occur prior to the first non-configuration access targeting the device.

Bit descriptions

The following image shows the higher register bit assignments.

Figure 5-306: lcn_hashed_target_grp_compact_hash_ctrl0-31

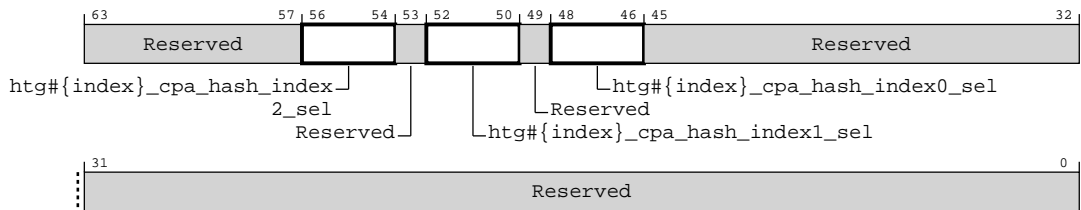


Table 5-310: lcn_hashed_target_grp_compact_hash_ctrl0-31 attributes

Bits	Name	Description	Type	Reset
[63:57]	Reserved	Reserved	RO	-
[56:54]	htg#{index}_cpa_hash_index2_sel	Configures the CPAG hash selection bits from the total hnfs hash across SMP. 3'b000: pass through from the SMP hnf_hash_index2. 3'b001: SMP hash index2 + 1. 3'b010: SMP hash index2 + 2. 3'b011: SMP hash index2 + 3. 3'b100: SMP hash index2 + 4. 3'b101: SMP hash index2 + 5. 3'b110: SMP hash index2 + 6. 3'b111: Hardcoded value (1'b0)	RW	3'b0
[53]	Reserved	Reserved	RO	-
[52:50]	htg#{index}_cpa_hash_index1_sel	Configures the CPAG hash selection bits from the total hnfs hash across SMP. 3'b000: pass through from the SMP hnf_hash_index1. 3'b001: SMP hash index1 + 1. 3'b010: SMP hash index1 + 2. 3'b011: SMP hash index1 + 3. 3'b100: SMP hash index1 + 4. 3'b101: SMP hash index1 + 5. 3'b110: SMP hash index1 + 6. 3'b111: Hardcoded value (1'b0)	RW	3'b0
[49]	Reserved	Reserved	RO	-
[48:46]	htg#{index}_cpa_hash_index0_sel	Configures the CPAG hash selection bits from the total hnfs hash across SMP. 3'b000: pass through from the SMP hnf_hash_index0. 3'b001: SMP hash index0 + 1. 3'b010: SMP hash index0 + 2. 3'b011: SMP hash index0 + 3. 3'b100: SMP hash index0 + 4. 3'b101: SMP hash index0 + 5. 3'b110: SMP hash index0 + 6. 3'b111: Hardcoded value (1'b0)	RW	3'b0
[45:0]	Reserved	Reserved	RO	-

Appendix A Revisions

Changes between released issues of this manual are summarized in tables.

The first table is for the first release. Then, each table compares the new issue of the manual with the last released issue of the manual. Release numbers match the revision history in [Release Information](#) on page 2.

Table A-1: Issue 0000-01

Change	Location
First release of documentation	-